Conceção e instalação de um controlador de um *spindle* para integração em célula robótica

Fábio André Fonseca de Castro

Dissertação do MIEM

Orientadores:
Prof. Paulo Augusto Ferreira de Abreu
Prof. António Pessoa de Magalhães

Faculdade de Engenharia da Universidade do Porto
Mestrado Integrado em Engenharia Mecânica

Julho 2012

"A mente que se abre a uma nova ideia jamais voltará ao seu tamanho original."
Albert Einstein

Resumo

Os campos de aplicação dos robôs industriais vão desde operações de simples manipulação, montagem, soldadura até operações de controlo de qualidade e maquinagem. Dentro destas operações de maquinagem, é necessário que o robô possua um motor árvore (ou, em inglês, *spindle*) para fornecer o movimento de rotação necessário à ferramenta de corte. É a instalação de um destes equipamentos num robô ABB IRB 2400 que é abordada nesta dissertação.

A arquitetura de controlo implementada utiliza um autómato, que comunica com um variador de frequencia para controlo da velocidade e sentido de rotação da ferramenta através de uma ligação RS485 e com o controlador ABB IRC 5 através de uma rede *Ethernet*.

Com vista à utilização segura do equipamento, a pressão do circuito pneumático, a temperatura no interior do *spindle*, a presença de ferramenta de corte e o estado dos botões de emergência são permanentemente monitorizados.

O controlador do *spindle* desenvolvido permite operar em dois modos: manual e automático. No modo manual, o operador comanda o spindle via consola HMI do autómato. No modo automático, o *spindle* é comandado via controlador do robô, tendo sido desenvolvidas rotinas de programação em linguagem RAPID para integração na programação do robô.

Abstract

Design and installation of a spindle's driver for integration into robotic cell

Typical applications of robots include welding, painting, assembly, pick and place (such as packaging, palletizing), product inspection and machining operations, all accomplished with high endurance, speed, and precision.

Machining robots must be extremely rigid and strong to be able to machine hard materials such as metal. A spindle must be attached to the end of the robotic arm in order to provide the rotation speed required for the cutting tool. This equipment must be flexible, durable and meet the payload limitations of the robot. In this project, we will discuss the installation of one of these devices in a robot ABB IRB 2400.

The spindle is controlled by a PLC, which communicates with an AC Motor Drive via a RS485 link and with an ABB IRC5 controller via an *Ethernet* network. The pressure of the pneumatic circuit, the temperature inside the spindle, the status of the cutting tool and the emergency buttons are controlled to ensure the safety of the system. The system can be controlled manually by HMI console or by the robot controller via RAPID routines.

Agradecimentos

Começo por expressar aqui um profundo agradecimento aos meus orientadores, Professor Paulo Abreu e Professor António Pessoa de Magalhães, pelo seu apoio, disponibilidade e incansável dedicação ao longo de todo este trabalho. Pelas longas horas que despenderam comigo na procura de soluções para os diversos problemas que surgiram com o desenrolar deste projecto, e pela partilha de conhecimentos que me ajudaram em várias fases a melhorar este projeto, deixo aqui o meu sincero Obrigado.

Agradeço tambem ao coordenador da opção de Automação do MIEM, Professor Francisco Freitas, pelo acompanhamento e criticas contrutivas feitas ao longo deste projeto.

Gostaria também de agradecer a ajuda do Assistente Técnico Eng. Joaquim Silva pela ajuda na montagem dos diversos componentes e dicas para resolução de problemas que foram surgindo.

A todos os meus colegas da opção de Automação, em especial ao Ruben Almeida, Sara Fernandes e Daniel Angelino, pela ajuda e conselhos que deram e pelos bons momentos de descontração e companheirismo.

À minha namorada Cátia Lima Silva por todos os momentos de felicidade, apoio e confiança que me dá.

Por fim gostaria de agradecer à minha família, em especial aos meus pais, que são um exemplo de vida para mim e sempre me apoiaram em todas as circunstâncias, fazendo-me perceber que nem sempre o caminho mais fácil é o mais proveitoso.

Índice

Lista de Figuras

Lista de Tabelas

Capítulo 1
Introdução

1.1 Introdução

Uma das aplicações típicas de robôs industriais diz respeito a operações de acabamento (rebarbagem, polimento, fresagem...) estando o robô equipado com um motor árvore (*spindle*). No *spindle* é montada uma ferramenta adequada à operação de acabamento pretendida. Tipicamente, o *spindle* é controlado a partir de um variador de frequência, devendo o seu comando estar integrado com o controlador do robô de modo a ser possível definir as condições de operação.

Os robôs equipados com *spindles* possuem geralmente um baixo custo, facilidade de programação e boa adaptabilidade e flexibilidade quando comparados com as máquinas CNC normalmente utilizadas para este tipo de aplicações. Como desvantagem, os robôs apresentam alguns problemas de precisão dimensional mas a evolução tecnológica destes equipamentos tem vindo a reduzir cada vez mais este tipo de problemas.

A utilização de robôs industriais equipados com *spindles* é também muito comum na área da prototipagem.

As diferentes tecnologias que irão ser utilizadas na implementação da arquitetura de controlo apresentada estão descritas com algum detalhe ao longo do segundo capítulo.

No terceiro capítulo, é apresentado em pormenor a arquitetura de controlo a implementar, bem como todos os componentes principais necessários à sua implementação. São também descritos todos os requisitos que o sistema deverá respeitar.

A implementação de todas as comunicações necessárias e descrição de todas funcionalidades implementadas vêm descritas no capítulo quarto.

Para finalizar, as conclusões da dissertação e as sugestões de trabalhos futuros são apresentadas no capítulo cinco.

1.2 Objetivos da dissertação

Este trabalho propõe a instalação de um *spindle* da marca PDS no robô industrial ABB IRB 2400 existente no laboratório de Robótica do Departamento de Engenharia Mecânica da Faculdade de Engenharia da Universidade do Porto.

A arquitetura de controlo do sistema a implementar passará pela utilização de um autómato programável que comunicará com o driver (variador de frequência) do *spindle* através de uma ligação série RS485 e com o controlador do robô através de uma rede *Ethernet*.

As necessidades de comando do *spindle* passam por definir a velocidade de operação, sentido de rotação, o comando (*on/off*) do sistema pneumático automático de mudança de ferramenta e o comando (*on/off*) do circuito de refrigeração a ar do *spindle*.

Na programação do autómato, deverão ser implementadas as funções de segurança necessárias à operação do *spindle*, o que requer a deteção da pressurização do circuito pneumático, da presença de ferramenta, do excesso de temperatura de funcionamento do *spindle* e da atuação dos botões da emergência instalados.

O sistema final deve possuir dois modos de funcionamento: o Funcionamento Manual e o Funcionamento Automático. No Funcionamento Manual, o *spindle* é controlado através da HMI integrada no autómato sendo que o utilizador indica os parâmetros de funcionamento. Já no Funcionamento Automático, toda a operação do *spindle* é controlada pelo controlador do robô.

Capítulo 2
Estado da Arte

2.1 Robôs Industriais

O termo *Robô* foi usado pela primeira vez pelo escritor checo Karel Capek (1890-1938) na peça de teatro “R.U.R. (*Rossum's Universal Robots*) ”, estreada em Janeiro de 1921 na cidade de Praga. A origem da palavra “robô” vem da palavra checa “robota” que significa trabalho forçado, daí a ideia que existe do robô que imita o homem em todas as atividades, funcionando como um “empregado” mecânico [1].
Já o termo Robótica foi popularizado pelo escritor e cientista americano Isaac Asimov, na sua obra "*I, Robot*" de 1950. Neste mesmo livro, Asimov criou três leis que, segundo ele, regeriam os robôs no futuro:

1. Um robô não pode fazer mal a um ser humano e nem, por omissão, permitir que algum mal lhe aconteça.
2. Um robô deve obedecer às ordens dos seres humanos, exceto quando estas contrariarem a Primeira lei.
3. Um robô deve proteger a sua integridade física, desde que, com isto, não contrarie a Primeira e a Segunda leis [2].

Neste livro, os robôs são desenhados com muitas semelhanças à espécie humana. Já existiam até desenhos de Leonardo Da Vinci para um projeto de um autómato humanoide que deverão

ter sido feitos por volta do ano de 1495. Este projeto continha desenhos detalhados de um cavaleiro mecânico que seria capaz de se sentar, mover a cabeça, maxilar e mexer os braços [3].

A ideia da construção de robôs surgiu no início do século XX com vista a aumentar a produtividade e a melhorar a qualidade dos produtos. Um robô industrial é oficialmente definido pela norma ISO como um "manipulador multitarefa controlado automaticamente, reprogramável, com movimento em três ou mais eixos" [4].

Os robôs têm vindo a ser utilizados numa gama muito variada de aplicações industriais. Estes equipamentos são especialmente utilizados em tarefas facilmente automatizáveis, como por exemplo algumas operações de carga e descarga, ou em ambientes onde as condições de trabalho sejam adversas para o ser humano (temperatura e ruído elevados, ambientes radioativos, presença de fumos ou poeiras, etc.) [5].

Em ambientes extremamente ruidosos, poeirentos e perigosos, a automatização do processo é altamente recomendada. Este facto é, assim, um impulsionador para o aumento da utilização de robôs em determinados processos industriais [5]. A figura 1 apresenta um gráfico da evolução do número de robôs fornecidos entre o período de 1998 a 2010.

Algumas das aplicações típicas dos robôs industriais incluem pintura, soldadura, montagem, movimentação de cargas, inspeção de produtos, e realização de testes de qualidade, tudo realizado com uma precisão, velocidade, e robustez relativamente elevadas [5].
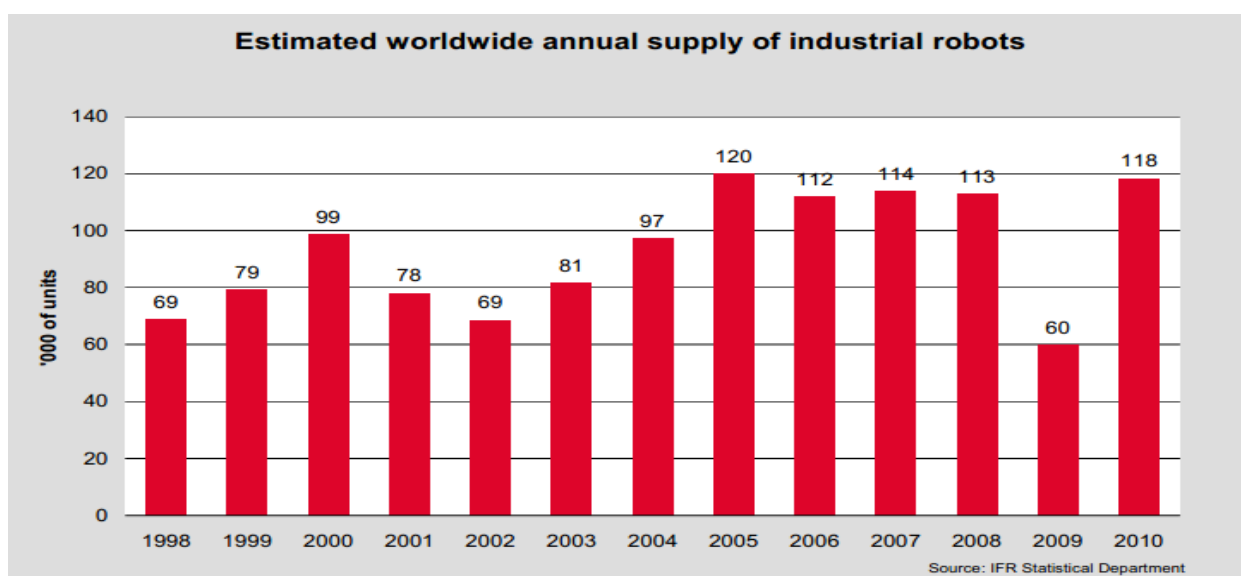
Figura 1 - Número de robôs industriais fornecidos em 2010 [6]

Em 2008, cerca de 70% dos robôs existentes são usados em operações de soldadura e de carga/descarga, como mostra a figura 2 [7].
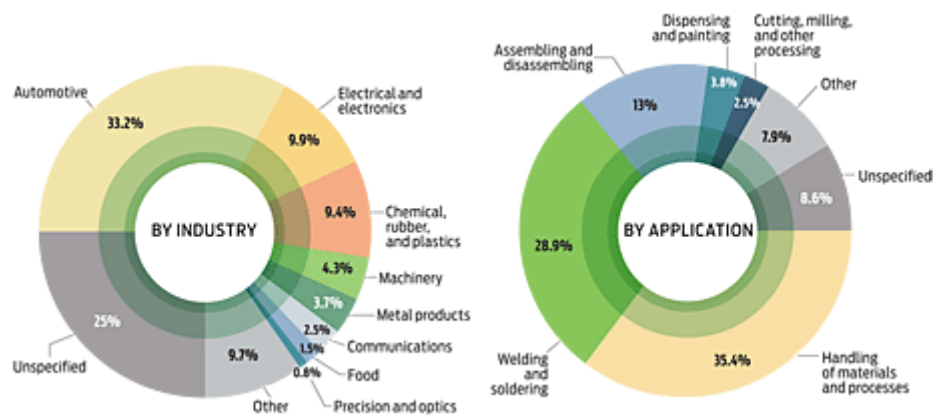
Figura 2 - Utilização de Robôs por tipo de indústria (à esquerda) e por aplicação (à direita) em 2008 [7]

2.2 Robôs Industriais em Operações de Maquinagem

Tipicamente, as operações de maquinagem são realizadas recorrendo a um centro de maquinagem CNC. Estas soluções de maquinagem CNC possuem custos elevados e alguma dificuldade de retorno de investimento [5].

É assim que surge a ideia da utilização de robôs industriais para realização de algumas operações de maquinagem onde os requisitos de precisão dimensional da peça não sejam muito elevados, uma vez que os robôs são bastante adaptáveis e flexíveis, fáceis de programar e possuem um custo relativamente baixo. Dentro das aplicações de maquinagem, os robôs industriais são usados predominantemente em aplicações de prototipagem, limpeza e pré-maquinagem de peças de fundição, assim como em operações de acabamento em aplicações onde a tolerância dimensional não é muito exigente [8]. Apesar disto, não existem muitos exemplos de sucesso da utilização de robôs industriais neste tipo de soluções a nível industrial [5].

O maior obstáculo à utilização de robôs neste tipo de aplicações é o facto de a rigidez de um robô industrial ser muito menor do que a de uma máquina CNC standard. A rigidez típica de um robô articulado é normalmente menor que 1N/µm enquanto uma máquina CNC possui uma rigidez por volta de 50N/µm [5].

Este facto leva a que a precisão dimensional do robô seja substancialmente menor do que uma máquina CNC, o que poderá ser um impedimento à sua utilização em algumas aplicações. Existem também problemas relacionados com vibrações no manipulador, problemas de precisão na programação *off-line* e calibração que dificultam a utilização de robôs neste campo [5].

Tem existido uma intensa investigação neste campo de forma a desenvolver estratégias para aumentar a precisão do robô e "compensar" o baixo valor da rigidez de um manipulador robótico. Estratégias como a utilização de um *Robot Stiffness Model* para compensar as deformações sofridas durante a maquinagem [5], a variação da velocidade de rotação do *spindle* para redução de vibrações [10], entre outros, foram elaboradas e demonstraram bons resultados, tornando cada vez mais viável a utilização de robôs neste tipo de aplicações. Uma vez que estas estratégias não pertencem ao âmbito desta dissertação, não serão aqui abordadas em mais detalhe.

Figura 3 - Exemplo de um Robô equipado com um s*pindle* numa operação de maquinagem [9]

2.3 Spindles

Para estas aplicações de maquinagem são normalmente utilizados *spindles* (ou motores árvore) montados no extremo do manipulador robótico, como é mostrado na figura 3, de modo a fornecer o movimento de rotação necessário à ferramenta de corte, bem como a potência necessária para a operação.

As características mais importantes para caracterizar um *spindle* são [11]:

a) Potencia Nominal e Binário
b) Máxima e Mínima Velocidade de Rotação
c) Tipo e Tamanho de Ferramenta a utilizar
d) Tipo de refrigeração
e) Peso

Existem basicamente dois tipos de *spindles*: os *spindles* com motor integrado (*Integrated Motor Spindle*) e os *spindles* acionados por correias (*Belt Driven Spindle*). A figura 4 apresenta um exemplo de cada um dos tipos.

Figura 4 - Um *spindle* com motor integrado (*Integrated Motor Spindle*) à esquerda e um *spindle* acionado por correia (*Belt Driven Spindle*) à direita [11, 12].

Os *spindle* acionados por correias são normalmente constituidos por um eixo apoiado num sistema de rolamentos que são introduzidos dentro de um corpo, normalmente metálico. O eixo do *spindle* incorpora tambem o sistema de fixação da ferramenta. O sistema que possiblita a mudança da ferramenta é, geralmente, montado externamente [11].
A potência e o momento de rotação são fornecidos ao *spindle* por meio de um motor externo. O motor transmite o binário usualmente usando correntes dentadas ou correias em V.
As principais vantagens deste tipo de *spindle*s são [11]:

- Custo reduzido: como o *spindle* é constituido por um número menor de componentes, o seu custo é reduzido quando comparado com os *spindle*s com motor integrado.
- Grande variedade possível de características: como a potência, o binário e a velocidade máxima são muito dependentes do motor externo, é possível variar os valores destas características simplesmente alterando este último, o que não é possível num *spindle* com motor integrado.
- Grande potência e binário disponível: uma vez que o motor é montado externamente ao eixo do *spindle*, é muitas vezes possível utilizar motores de grandes dimensões. Motores de grandes dimensões possuem habitualmente grandes potências e binários que podem ser transmitidos ao *spindle*. No caso dos *spindles* com motor integrado, o espaço livre é limitado, o que impede o uso de motores de grandes dimensões.

Este tipo de *spindle*s estão geralmente limitados a uma velocidade máxima entre 12000 e 15000 rotações por minuto e podem encontrar-se no mercado com potências até cerca de 30HP [11].

Nos *spindles* com motor integrado, o motor é montado dentro da estrutura e é acoplado ao eixo do *spindle*. Isto permite que o *spindle* atinga velocidades elevadas sem problemas adicionais relacionados com os esforços induzidos pelas correias ou rodas dentadas. O eixo do *spindle* é posicionado por um conjunto de rolamentos de precisão. Estes rolamentos podem necessitar de manutenção frequente [11].

Neste caso, as características do *spindle* estão diretamente ligadas às do motor. A potência e velocidades máximas do *spindle* estão limitadas às características do motor, não sendo geralmente possível a substituição deste por um de características diferentes. Assim, a motor utilizado é um dos critérios mais importantes neste tipo de equipamento [11].

Nestes equipamentos é necessário providenciar o arrefecimento do motor interno. Este arrefecimento é efetuado geralmente por ar ou água.

As principais vantagens deste equipamento em comparação com os *spindle*s acionados por correia são [11]:

- Velocidades máximas mais elevadas: a utilização de correias ou correntes traz alguns problemas associados. Em primeiro lugar, os problemas de escorregamento das correias impossibilitam que se use toda a energia fornecida. Para além disso, as temperaturas geradas pelo atrito de contacto da correia com o eixo do *spindle* a elevadas velocidades pode danificar as correias. A utilização de rodas dentadas e correntes elimina estes problemas mas induz vibrações prejudiciais ao tipo de operação que estes equipamentos realizam. Nos *spindle* com motor integrado, o motor é acoplado diretamente ao eixo do *spindle*, eliminando os problemas anteriormente apresentados e permitindo que velocidades superiores sejam atingidas.
- Esforços menores nos rolamentos: as correias necessitam de estar tensionadas para funcionarem. Este tensionamento vai exercer uma força radial nos rolamentos onde o eixo do *spindle* está apoiado. Esta força radial aumenta com a potência e velocidade transmitida, o que poderá danificar os rolamentos se velocidades ou potências demasiado elevadas forem atingidas. Nos *spindle* com motor acoplado, o acoplamento do motor ao eixo não irá exercer forças radiais significativas nos rolamentos. Este facto pode reduzir a necessidade de manutenções frequentes.

Com vista ao correto funcionamento e longevidade destes *spindle*s, existem alguns parâmetros de segurança que tem de ser, obrigatoriamente, tidos em conta. A temperatura no interior do *spindle* (no caso dos *spindle* com motor integrado) e o bloqueio/desbloqueio da ferramenta são apenas dois dos parâmetros que têm de ser monitorizados durante o funcionamento do equipamento. Durante as operações de manipulação, manutenção e reparação do *spindle* é indispensável que o *spindle* esteja parado e, preferencialmente, desconectado da corrente elétrica [11].

Existem *spindles* com uma potência desde 100 W até 35 KW e com velocidades de rotação até cerca de 50000 rpm. Existem ainda *spindle*s especiais para maquinagem de pedra ou vidro [14].

2.4 Autómatos programáveis

Os autómatos programáveis, vulgarmente conhecidos por PLC (*Programmable Logic Controllers*), são elementos fundamentais nos sistemas modernos de automação. Estes equipamentos podem desempenhar diversas funções, como por exemplo coordenação geral do sistema, aquisição e processamento de dados, gestão de diversos tipos de comunicações, controlo local de baixo nível de vários sistemas, entre muitos outros.

O primeiro PLC foi inventado por Dick Morley em 1969. Nesta altura, os PLC surgiram como alternativa mais flexível à logica elétrica, eliminando a necessidade de substituição de *hardware* para cada nova configuração lógica, incrementando drasticamente as funcionalidades e reduzindo o espaço de montagem [16].

Todos os PLC, desde os mais elementares aos mais potentes, possuem algumas capacidades elementares como:

- *Timers*: permitem a implementação de funções de temporização.
- Contadores: contar eventos é uma das atividades mais comuns na sua utilização. Na maioria dos PLC existem contadores de alta velocidade para utilizações mais exigentes.
- Registos: permitem a escrita e consulta de dados em memória. Podem guardar desde bits a *double words*. A capacidade de memória aumenta muito com a gama do PLC.
- Operações lógicas elementares: permitem manipular informação e construir sequências lógicas de instruções essenciais ao seu funcionamento.

- Entradas e Saídas Digitais e Analógicas: às entradas podem ser ligados, entre muitos outros dispositivos, comutadores, botões, sensores de proximidade, pressostatos, *encoders*, sondas de temperatura, etc. As saídas são utilizadas para atuar válvulas, motores, atuadores pneumáticos ou hidráulicos, lâmpadas, relés, etc. Alguns PLC tem um esquema modular, permitindo a expansão de I/O com módulos dedicados a funções específicas.
- Possibilidade de ligação remota: de modo geral, todos os PLC permitem comunicações por ligações ponto-a-ponto, normalmente por RS232 ou RS485. Os mais avançados, podem suportar redes *Ethernet* baseadas em TCP/IP, redes CAN e profibus, etc.
- Funções matemáticas: muito úteis para cálculo de determinadas variáveis. O número das possíveis funções matemáticas depende muito da gama do PLC utilizado.
- Funções avançadas de controlo: por exemplo, blocos PID já construídos e configuráveis pelo utilizador é uma das funções normalmente existente nos PLC de gama média e alta [14, 15].

Atualmente, os PLC apresentam velocidades de execução da ordem de 0,5 a 1.8 µs por instrução, o que é uma performance bastante considerável [15]. A grande maioria dos PLC possui a estrutura representada na figura 5.
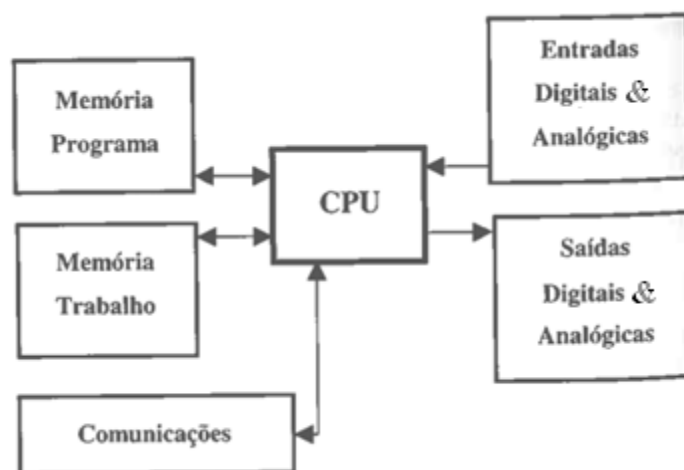
Figura 5 - Estrutura característica de um PLC [15]

A CPU é o elemento central do equipamento. É ela que monitoriza as entradas e atualiza as saídas de acordo com a programação efetuada. É também responsável por todas as operações matemáticas, por gerir *timers*, contadores e outras funções especiais. A memória do programa

é a zona de memória destinada a conter o programa em execução, sendo que os registos atualizados das entradas e saídas, o registo das operações matemáticas, lógicas e registos do utilizador são guardados na memória de trabalho [15].

Os PLC funcionam normalmente em 4 etapas, que estão apresentadas na figura 6. Estas etapas são continuamente repetidas em *loop* durante o funcionamento do equipamento.

Figura 6 - Ciclo de funcionamento de um PLC [16]

Geralmente a programação do PLC pode ser efetuada (segunda a norma IRC 61131) em [16]:

- ❖ Diagrama de Blocos Funcionais (FBD)
- ❖ Texto Estruturado (ST)
- ❖ Lista de Instruções/Booleana (IL)
- ❖ Diagrama *Ladder* (LD)
- ❖ GRAFCET (SFC)

Os PLC de gama baixa não suportam normalmente todas estas linguagens. As mais comuns de serem encontradas são a programação por diagrama *Ladder* (LD) ou por blocos funcionais (FBD).

Hoje em dia existem vários tipos de PLC, dimensionados para diversos tipos de aplicações, sendo os mais poderosos verdadeiros computadores industriais com elevada performance.

2.5 Meios de comunicação em ambientes industriais

Em todas as comunicações é necessário existir um transmissor e pelo menos um recetor. O principal objetivo é permitir o envio de informação do transmissor para um ou vários recetores. Assim é necessário existir um meio que lhes permita comunicar entre si. A comunicação pode ser realizada por condutores elétricos, cabos de fibra ótica, radiação eletromagnética (redes wireless), entre outros.

A figura 7 ilustra os 3 modos de comunicação normalmente utilizados em comunicações.
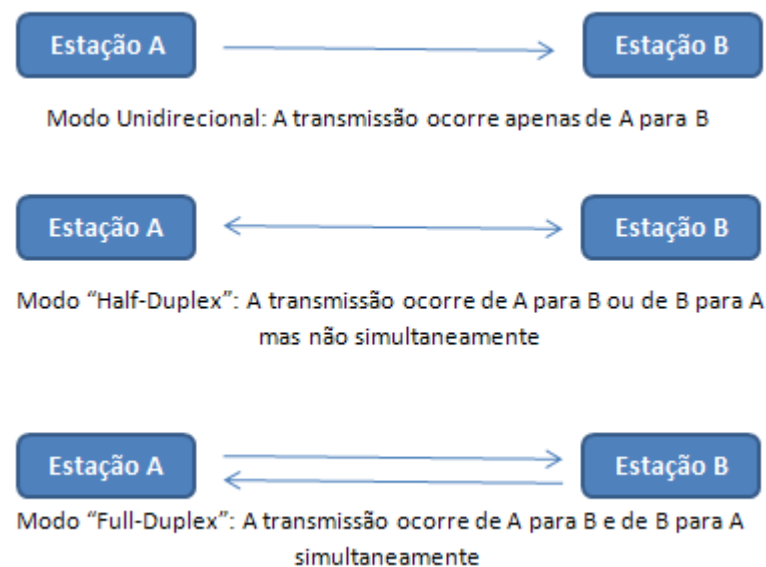
Figura 7 - Modos de comunicação utilizados [15]

No modo unidirecional, a transmissão de dados só se processa num sentido. No caso da figura 7, a transmissão dos dados ocorre da estação A para a estação B. No modo "*Half-Duplex*" a transmissão dos dados pode ocorrer em ambos os sentidos, mas nunca ocorre simultaneamente em ambos os sentidos. No modo "*Full-Duplex*", a transmissão dos dados pode ocorrer em ambos os sentidos simultaneamente [17].

Para haver comunicação é sempre necessário [17]:

- Um sistema de transmissão (hardware capaz de gerar e transmitir sinais através de um meio).
- Software de comunicação (software de gestão do hardware que lhe diz como e quando transmitir).
- Protocolos (acordos entre os intervenientes que gerem a transmissão, receção e interpretação da informação).

Existem, basicamente, dois tipos de comunicação entre diferentes equipamentos: a comunicação série e a comunicação paralela.

A comunicação em série permite transmitir apenas um 'bit' de cada vez. Assim a mensagem a transmitir tem de ser dividida em palavras, e de seguida em bits, sendo estes enviados individualmente como é mostrada na figura 8. Utiliza normalmente um canal de transmissão de dados, que pode ser por exemplo uma linha de telefone normal [17].
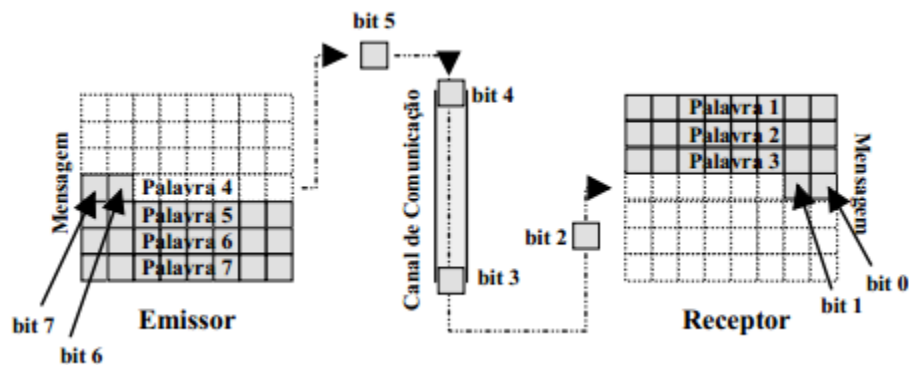
Figura 8 - Comunicação Série [17]

Já a comunicação paralela conta com um canal de comunicação que permite transmitir vários 'bits' (normalmente 7 ou 8 bits) em simultâneo (em paralelo) como é representado na figura 9. Utiliza vários condutores para transmissão de dados [17].
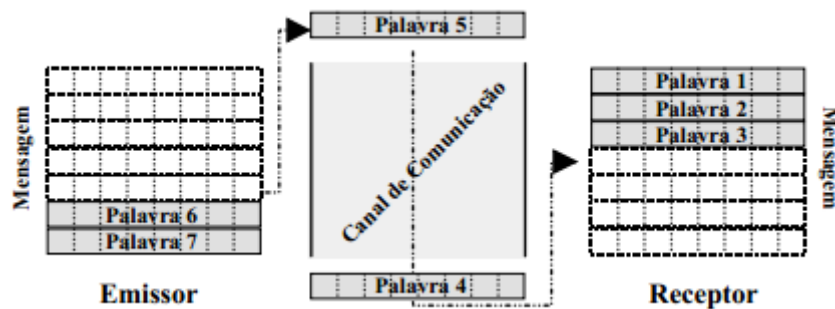
Figura 9 - Comunicação Paralela [17]

Como vantagens a comunicação paralela é rápida, simples (a nível de *hardware* e *software*) e boa para pequenas distâncias. Já a comunicação em série possui um baixo custo, é adaptada a canais telefónicos e hertzianos e, por isso, é boa para comunicações a grandes distâncias.

A comunicação paralela é muito utilizada para ligar periféricos às portas do hardware central. Os controladores de ecrã, de discos de um computador, são vulgarmente ligados ao *hardware* central por uma comunicação em paralelo, maximizando-se assim a velocidade de transferência [17].

Também os equipamentos periféricos que estão próximos de um computador são muitas vezes ligados ao *hardware* central por meio de interfaces paralelas. Porém, para maiores distâncias, este tipo de comunicação não é utilizado, uma vez que se torna bastante cara e porque o ruído existente não permite garantir a transmissão da informação sem erros. Estas questões são resolvidas com a comunicação série, a qual tem contudo o inconveniente de ser muito mais lenta e de apresentar maiores dificuldades a nível da implementação [17].
Existem dois modos de comunicação série: o modo síncrono e o modo assíncrono. Ambos efetuam processos de sincronização antes de iniciarem a transmissão de uma dada mensagem. Porém, no modo assíncrono, a sincronização é feita por um ‘ bit’, ao passo que no síncrono é feita por um byte.
A quantidade de informação enviada após uma sincronização varia bastante da comunicação síncrona para a assíncrona: na comunicação assíncrona, o número de bits enviado após uma sincronização é, tipicamente, 5 a 8 bits; na comunicação síncrona esse número é muito maior – poderão ser dezenas ou centenas de ‘ bits’ [17].
Dada a complexidade inerente nas comunicações, que podem envolver diferentes dispositivos, os quais podem ser de diferentes marcas, foram criados alguns modelos de forma a orientar o processo de implementação de uma rede de comunicação. Existem diversos modelos que estão orientados para áreas de aplicação específicas. Os mais utilizados são o modelo OSI e o modelo IEEE 802 LAN.

2.6 Modelo OSI

A utilização de normas nas comunicações de dados é uma necessidade óbvia. Estas são necessárias para gerir o uso e interligação de equipamentos tanto a nível físico, como elétrico e mesmo a nível dos processos e procedimentos manipulação os dados.
Assim, em 1978, é proposto pela *International Organization for Standardization* (ISO) o modelo OSI numa tentativa de implementar um *standard* na comunicação entre computadores. Não especifica a utilização de um equipamento específico (modem, tipo de ligação, etc.). O modelo divide as redes de computadores em sete níveis (ou camadas), de forma a obter níveis de abstração. Cada camada tem as suas funções, tem características específicas e é usada para executar determinadas tarefas. Cada camada possui uma estrutura própria e trata de um problema particular, sem se preocupar com a estrutura das outras camadas. Cada camada apenas terá de receber dados de um determinado tipo, processa-los e enviar num determinado formato para o próximo *layer*.

Figura 10 - Sete Camadas do Modelo OSI [18]

As camadas do modelo OSI apresentam as seguintes funções:

- Física (*Physical Layer*): controla a transmissão dos bits através do meio físico. Tipicamente, há definição das tensões associadas aos “uns” e “zeros” e a duração de um bit.
- Ligação de Dados *(Data Link Layer)* : controla o acesso ao meio, o fluxo de dados e a deteção e correção de erros.
- Rede (*Network Layer*): gere o encaminhamento dos dados, quando estes fluem entre diferentes redes.
- Transporte (*Transport Layer*): isola os níveis 1, 2 e 3, dos níveis 5, 6 e 7. Recebe mensagens do nível 5, divide-as em pacotes e passa-as para o nível 3 (ou vice-versa). Pretende garantir que as mensagens são trocadas correta e eficientemente.
- Sessão (*Session Layer*): gere os diálogos entre utilizadores.
- Apresentação (*Presentation Layer*): modificação de dados para compatibilização das aplicações. Exemplos: conversão de ficheiros, encriptação, compressão.
- Aplicação (*Aplication Layer*): proporciona serviços de rede às aplicações, caso os haja. [17]

A elaboração do modelo OSI representou um esforço na tentativa de padronização das novas tecnologias para que a implementação de produtos de redes fossem compatíveis entre si. Não existe nenhum protocolo que cubra completamente todas as camadas do modelo OSI. Existem protocolos que cobrem um ou mais *layers* do modelo e que, conjugados com outros protocolos que cobrem outros *layers*, permitem que a comunicação entre os equipamentos

seja possível e segura. A figura 10 apresenta alguns exemplos de protocolos e os respetivos layers do modelo OSI.

Figura 11- Camadas do modelo OSI com exemplo de protocolos [18]

2.7 RS-485

Uma das mais populares tecnologias para ligar equipamentos numa rede é o RS-485 (também conhecido por EIA-485).

O RS-485 é baseado na transmissão diferencial de dados, o que o torna ideal para transmissão em altas velocidades, longas distâncias e em ambientes propícios a interferência eletromagnética. Utiliza cabos com um ou dois pares de fios entrançados como meio de transmissão e é possível conectar até 32 dispositivos (com carga unitária) na mesma rede. Hoje em dia existem no mercado dispositivos com carga unitária inferior a um e, assim, é possível aumentar o número de dispositivos ligados à mesma rede desde que a carga unitária total dos dispositivos seja igual ou inferior a 32.
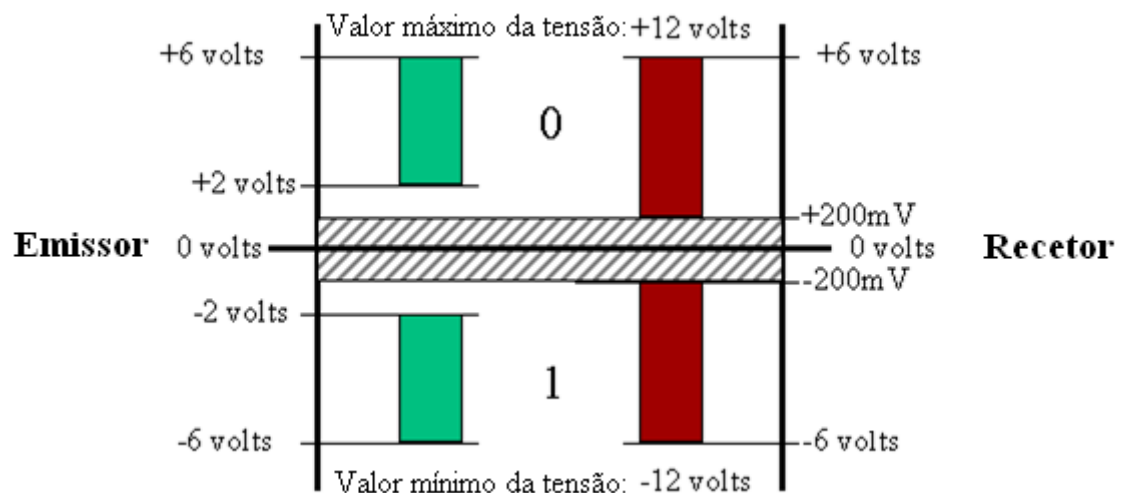
Figura 12- Valores de Tensão correspondentes aos valores lógicos "0" e "1" [19]

Os circuitos emissores e recetores utilizados utilizam como informação a diferença entre os níveis de tensão em cada condutor do par entrançado. Os códigos binários são identificados pela polaridade (+ ou -) da diferença de tensão entre os condutores do par. Quando a tensão no condutor "+" for maior que no condutor "-", é caracterizado um nível lógico "1". Quando, ao contrário, a tensão no condutor "-" for maior que no condutor "+", é caracterizado um nível lógico "0". Uma margem de ruído de ±0,2 V é definida para aumentar a tolerância a interferências. O facto de se utilizar pares de condutores entrançados resulta no cancelamento dos ruídos induzidos no meio de transmissão, uma vez que se o mesmo ruído é induzido nos 2 condutores, a diferença de tensão entre eles não se altera e assim os dados mantêm-se inalterados [20].
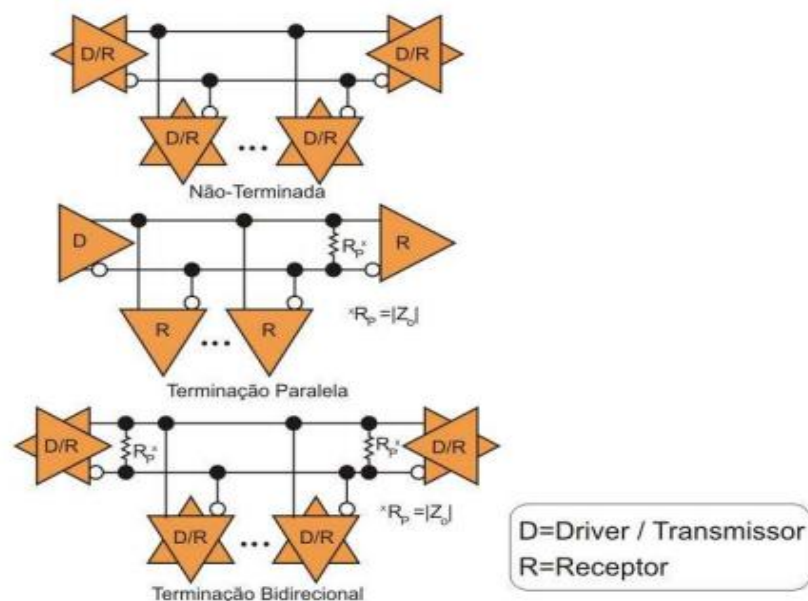
Figura 13 - Diferentes tipos de terminação que podem ser utilizados [21]

Assumindo que o cabo de transmissão é suficientemente longo, podem começar a surgir problemas de reflexão dos sinais. Para minimizar este problema, uma das soluções passa pela necessidade de usar terminação nas linhas de comunicação com um valor de impedância correspondente à impedância característica da linha de transmissão. Uma correta terminação atenua reflexões que distorcem os dados transmitidos, aumentando os limites de velocidade e/ou comprimento da rede. Alguns métodos de terminação são apresentados na figura 13 [21].
Redes não-terminadas são baratas, de menor consumo e simples de implementar. A desvantagem é que as taxas de comunicação devem ser lentas ou os cabos curtos o suficiente para manter a rede confiável. Redes com cabos curtos (até 100 m) e que operem a baixa velocidade funcionam adequadamente mesmo sem a utilização de resistências de terminação [21].
A terminação paralela oferece excelentes taxas de comunicação, mas é limitada a redes com um único *master*, onde um dispositivo fala e os demais apenas escutam. Nesses casos, o master deve ser posicionado numa extremidade da rede e a resistência de terminação na outra.
O último método representado é a terminação bidirecional, que permite uma excelente integridade do sinal. Com esta técnica, os masters podem estar localizados em qualquer ponto da rede mas tem a desvantagem de aumentar o consumo da rede. Este é, seguramente, um dos métodos mais confiáveis de terminação [21].
A impedância característica de um par trançado é de aproximadamente 120 ohms, sendo este um valor adequado para a resistência de terminação a ser instalado [21].

O RS-485 apresenta uma velocidade máxima de 10 Mbps com comprimentos de cabo inferiores a 10 metros e tem um alcance máximo de 1200 metros. A figura 14 apresenta um gráfico da evolução da velocidade máxima com o aumento do comprimento do cabo. Estas velocidades podem ainda ser afetadas por outros fatores externos como por exemplo características dos equipamentos instalados, capacidade dos cabos de comunicação, topologia da rede instalada e interferências presentes no ambiente de comunicação [20].
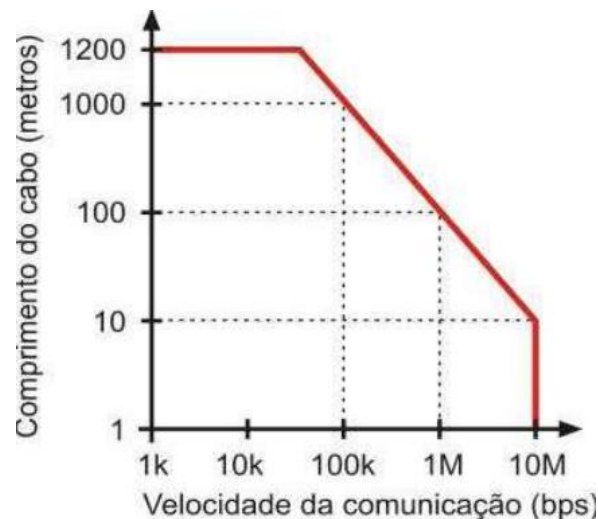
Figura 14 - Velocidade máxima usando EIA485 em relação ao comprimento do cabo [21]

Enquanto os comprimentos de cabo são relativamente curtas, a influência da topologia da rede no seu desempenho não é significativa. Contudo, quando os efeitos de linhas de transmissão longas começam a aparecer, há uma topologia simples que permite corrigir os problemas que daí advêm. A figura 15 mostra alguns tipos de topologias utilizadas. Apenas no tipo "*daisy chain*", onde todos os dispositivos são conectados diretamente aos condutores da linha de comunicação principal, é fácil controlar as reflexões causadoras de erros de comunicação.
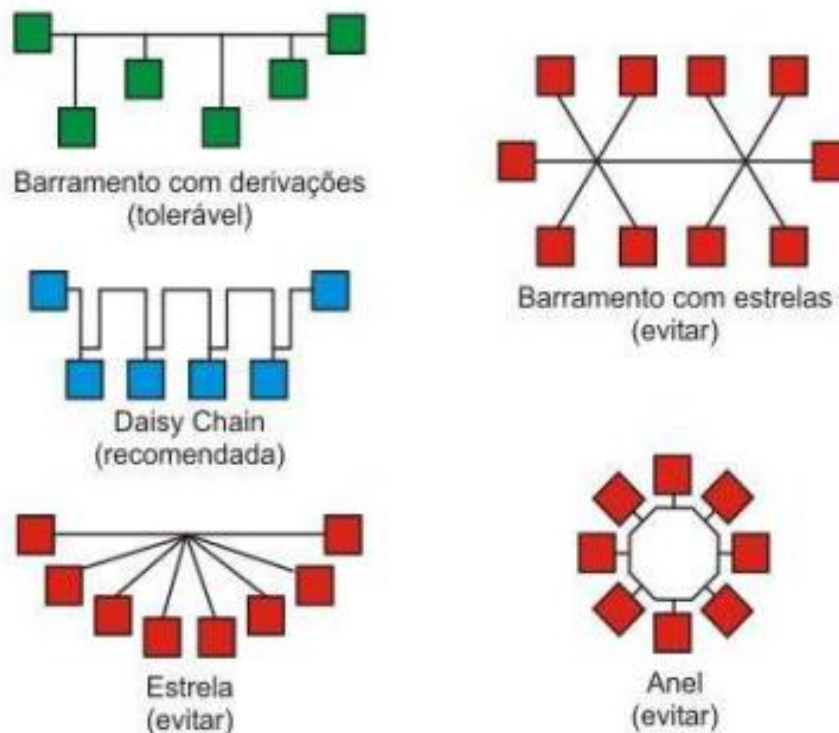
Figura 15 - Diferentes topologias de rede que podem ser usadas com EIA 485 [21]

Numa rede RS485 deve ser também instalado um terceiro condutor de forma a garantir o equilíbrio de tensão entre os diversos dispositivos da rede. Caso o condutor comum não seja instalado entre todos os dispositivos, todos os dispositivos devem ser adequadamente ligados a terra, de forma a garantir a compatibilidade elétrica de todos os equipamentos [21]

Existem dois tipos de redes que podem ser implementadas: RS-485 *half-duplex* (2 fios) ou RS-485 *full-duplex* (4 fios).
O tipo RS-485 *half-duplex* é a configuração mais usual. Utiliza um único par de fios para transmissão e receção de dados. Múltiplos dispositivos são ligados na forma de um barramento, conforme ilustra a figura 16.
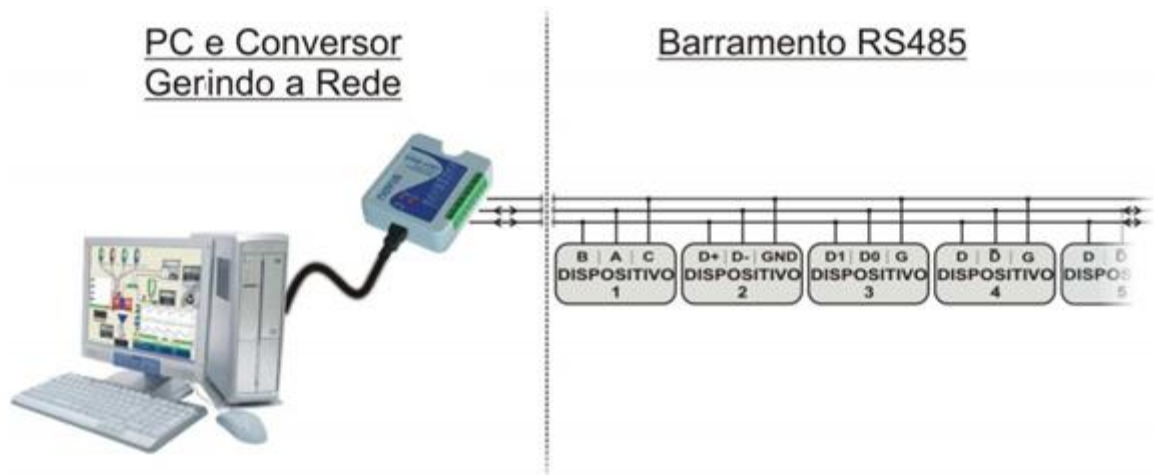
Figura 16 - RS 485 *Half Duplex* (a 2 fios) [20]

O RS-485 *full-duplex* utiliza dois pares de fios para a comunicação. A imagem 17 apresenta um exemplo de uma rede deste tipo. Por um par de fios fluem os dados transmitidos no sentido Conversor→Dispositivos da rede (par de transmissão do conversor) e pelo outro par os dados transmitidos no sentido Dispositivos da Rede→Conversor (par de receção do conversor) [20].
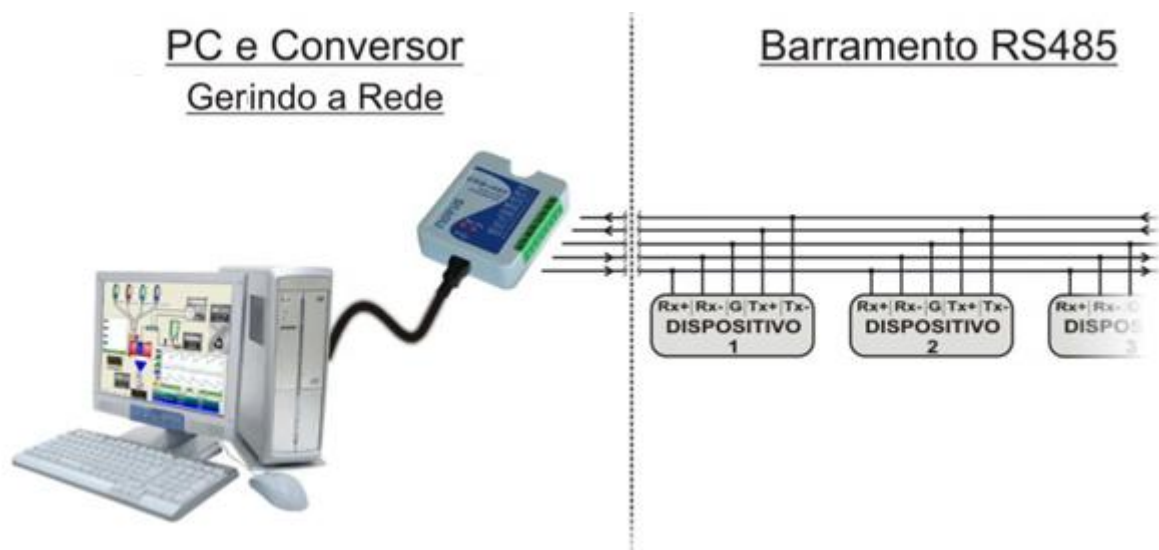
Figura 17 - RS 485 *Full Duplex* (a 4 fios) [20]

2.8 Ethernet

Ethernet é a tecnologia LAN (*Local Area Network*) mais utilizada nos dias de hoje.

A *Ethernet* surge em 1973 quando Bob Metcalfe escreve um artigo descrevendo como o sistema *Ethernet* por ele inventado que permitia conectar diferentes estações de trabalho, sendo possível enviar dados de uma estação para a outra e ainda para impressoras.

O sistema desenvolvido baseava-se num sistema existente na altura, o sistema Aloha. Este sistema tinha sido desenvolvido na Universidade do Havai no final dos anos 60 para ser utilizado em comunicações de rádio entre as diferentes ilhas existentes no arquipélago [22].

O algoritmo usado pelo sistema Aloha é bastante simples: qualquer estação pode enviar dados quando mais lhe convier, e depois aguarda a confirmação da receção dos dados pela estação de destino. Se não receber nenhuma confirmação num espaço relativamente curto de tempo, a estação assume que outra estação estava a transmitir ao mesmo tempo e, assim, ocorreu uma colisão na transmissão dos dados. Assim a estação que deveria receber a mensagem não recebeu nenhuma das mensagens em condições e não enviou a confirmação da receção dos dados. Assim, as estações que estavam a enviar os dados deveriam escolher um tempo de paragem aleatório e, após esse tempo, tentar enviar os dados novamente [22].

Uma das desvantagens deste sistema era que, quando o tráfego fosse intenso, o número de colisões iria aumentar exponencialmente, o que iria tornar morosa a transmissão dos dados.

Assim, Bob Metcalfe percebeu que poderia melhorar o sistema Aloha, introduzindo alguns melhoramentos. Para isso desenvolveu um novo sistema que possuía um mecanismo para deteção de colisões durante a comunicação (*collision detect*). O sistema incluía também a estratégia "*listen before talk*", segundo a qual as estações deviam verificar se existia atividade no meio (*carrier sense*) antes de iniciarem a transmissão dos seus dados. Este sistema permitia também o acesso á rede por várias estações (*multiple access*). Assim surge o protocolo de acesso ao meio utilizado na *Ethernet*, o *Carrier Senser Multiple Access with Collision Detect (CSMA/CD)*. Juntamente com um algoritmo de "*back-off*" mais sofisticado, a *Ethernet* poderia funcionar com 100% de tempo de utilização do meio. O protocolo de acesso ao meio usado na *Ethernet*, de forma simplificada, é apresentado na figura 18 [22].
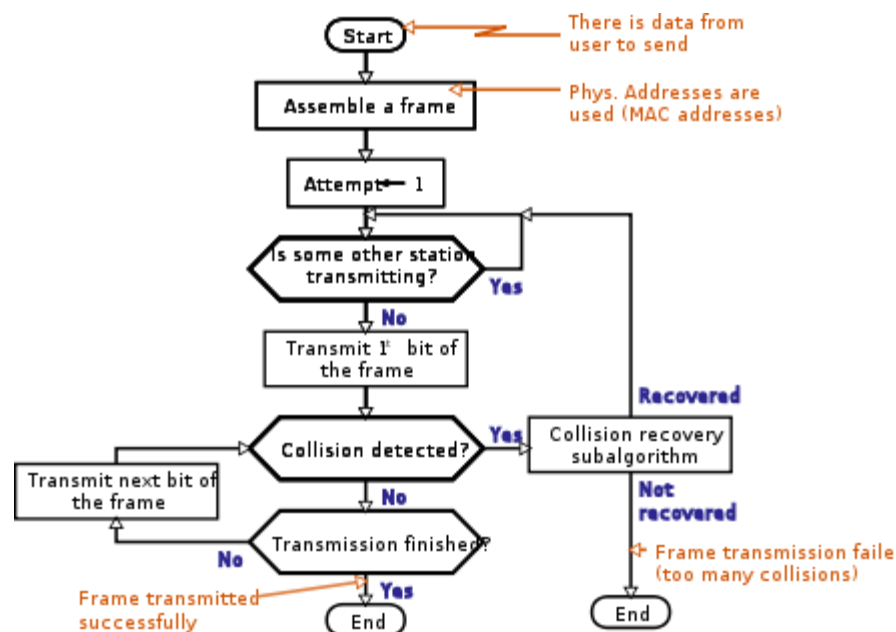
Figura 18 - Protocolo de Acesso ao meio utilizado na *Ethernet* (CSMA/CD) [23]

A primeira experiência usando o sistema *Ethernet* foi realizada no final de 1972 na comunicação entre dois computadores Xenox Alto e foi atingida uma velocidade de transmissão de dados de 2.94 Mbps. Desde esta altura, já foram desenvolvidos diversas evoluções ao sistema desenvolvido por Bob Metcalfe.

A *Ethernet* trata das duas últimas camadas do modelo OSI, ou seja, da camada física (*physical layer*) e camada de ligação lógica (*data link layer*).

Existem neste momento 3 tipos diferentes de *Ethernet* no que se relaciona com camada física do modelo OSI:

- 10Mbps *Ethernet*

- 100 Mbps *Ethernet* (ou Fast *Ethernet*)
- 1000Mbps *Ethernet* (ou Gigabit *Ethernet*)

Cada uma destas evoluções trouxe consigo uma forte evolução na velocidade e diversidade de aplicações em que pode ser utilizada. O formato da mensagem a enviar permanece o mesmo em todas as evoluções. A migração entre as diferentes evoluções é assim bastante fácil e barata, uma vez que praticamente todas as especificações técnicas são mantidas inalteradas.

A 10 Mbps *Ethernet* surge no início dos anos 90. Dentro deste sistema, surgiram duas variantes importantes [22]:

- 10 Base-T: utiliza 2 pares de condutores entrelaçados (sendo que um fio transposta o sinal positivo (0 a 2,5V) e outro o sinal negativo (0 a -2,5V), tendo um alcance de aproximadamente 100 metros. Nesta configuração, normalmente cada estação de trabalho é ligado a um *hub* ou *switch*. Estes equipamentos permitem gerir o processo de transmissão da informação e enviá-la para vários recetores ao mesmo tempo.
- 10 Base-F: utiliza fibras óticas e tem um alcance de aproximadamente 2 quilómetros. A arquitetura normal neste tipo de rede apresenta uma estrutura em estrela, na qual todas as estações estão ligadas a um *Hub/Switch* central.

A *Fast Ethernet* surge por volta de 1995 e conseguiu elevar a velocidade de transmissão de dados para 100 Mbit/s. A *Fast Ethernet* garante compatibilidade com as redes *Ethernet* anteriormente utilizadas e apresenta uma relação custo/benefício muito satisfatória, sendo por isso a mais utilizada. Possui dois padrões bastante difundidos que são [22]:

- 100 Base-TX: utiliza 2 pares de condutores entrelaçados e é o mais utilizado dentro da *Fast Ethernet*
- 100 Base-FX: é uma evolução da 10 Base-F, utiliza também fibras óticas com uma velocidade 10 vezes. Pode ter um alcance até 2000 metros.

A Gigabit *Ethernet* possui uma velocidade de transmissão de 1 Gbps. Existem atualmente várias tecnologias diferentes dentro da Gigabit *Ethernet*, sendo as mais importantes [22]:

- 1000 Base-T: pode ser utilizada em redes com uma distância inferior a 100 metros. Utiliza cabos com pares entrelaçados (como as redes de Fast *Ethernet*) e usa 4 pares de fios disponíveis.

- 1000 Base-CX: é o padrão inicialmente desenvolvido na Gigabit *Ethernet*. Utiliza cabos de pares entrelaçados como meio de transmissão e tem um alcance até 25 metros. Devido a esta limitação de distância, é a menos usada dentro da Gigabit *Ethernet*, sendo dificilmente encontrada em aplicações nos dias de hoje.
- 1000 Base-SX: esta tecnologia utiliza fibras óticas e é recomendada para distâncias até 550 metros. Possui um custo mais baixo do que a 1000 Base-LX, que utiliza também fibras óticas.
- 1000 Base-LX: esta tecnologia é a mais cara das aqui apresentadas, utiliza também fibras óticas como meio de transmissão e possui um alcance máximo de 5 km.

Um outro ponto fulcral para o sucesso das comunicações é a estrutura da mensagem a enviar aos diversos dispositivos.

Inicialmente, a estrutura da mensagem utilizada era *standard* DIX (DEC-Intel-Xerox), sendo mais tarde alterada para o standard IEEE 802.3, sendo este último atualmente o formato oficial. Esta estrutura é apresentada na figura 19.

Figura 19 - Estrutura da Mensagem segundo o standard IEEE 802.3 utilizada no protocolo *Ethernet* [24]

Preamble + SOF **–** Serve para sincronização do hardware existente na rede. É utilizado no sistema de 10 Mbps *Ethernet*. O *preamble* existe para permitir que os primeiros bits da mensagem possam ser perdidos devido aos *delays* que possam acontecer nos equipamentos. Os sistemas mais recentes de 100 Mbps e 1000 Mbps utilizam mecanismos mais complexos para codificação dos sinais, o que elimina a necessidade de um preâmbulo. Ainda assim, mesmo nestes sistemas, o preâmbulo é transmitido de forma a que a estrutura permaneça a mesma em todos os sistemas *Ethernet* [22].

Destination Address Field e Source Address Field **-** é o endereço do destinatário e do remetente da mensagem, respetivamente.

Length/ Type Field **-** Se neste campo o valor apresentado for igual ou inferior a 1500, este campo funciona como *length field*. Assim, indica o número de *bytes* de dados que irão ser

transmitidos de seguida. Se o valor for superior a 1536, este campo funciona como *type field*. Assim, neste campo é identificado o protocolo que foi utilizado [22].

Data Field **–** contém os dados a ser enviados, que podem variar o seu número entre 46 a 1500 bytes.

FCS (Frame Check Sequence) – também chamado de *Cyclic Redundancy Check (CRC).* Este campo contém valores utilizados para verificação da integridade dos dados. Neste campo, são enviados os coeficientes de um polinómio que são calculados utilizando os dados do *Destination Adress Field*, *Source Adress Field*, *Length/Type Field* e *Data Field*, não sendo utilizados dados do *Preamble* ou *SOF Field* [22].

2.9 Protocolo ModBus

Criado no ano de 1979 pelo fabricante de PLC's Modicon (atualmente uma marca da Schneider Electric's Telemecanique), o Modbus tinha como finalidade a comunicação entre nós duma rede *multi-drop* baseada numa arquitetura *master/slave*. Rapidamente, o Modbus foi implementado nos equipamentos de diversos fabricantes, uma vez que sendo um protocolo aberto, evita a necessidade de compra de licenças de outros protocolos [25].

O Modbus original corria através de RS232, mas posteriores implementações suportam também RS485, sendo uma vantagem uma vez que permite maiores distâncias, velocidades de comunicação e a possibilidade de usar uma rede *multi-drop*. É utilizado em redes com arquitetura *Master/Slave*. Este protocolo, que foi inicialmente usado em linhas de comunicação série, atualmente é também usado em comunicações sem fio e redes TCP/IP [26].

A comunicação em Modbus não se prende apenas com dispositivos inteligentes, como micro controladores, PLC's, etc. Também existem sensores equipados com um interface Modbus de forma a poderem transferir informação aos sistemas anfitriões.

O Modbus é muito utilizado para a transmissão de sinais de sensores, atuadores e outros instrumentos de medição para o controlador central do sistema. Por exemplo um sistema de medição de temperatura e humidade envia os resultados medidos para um computador central. É também bastante usado em aplicações com sistemas SCADA's (*supervisory control and data acquisition)* [25].

Através de interfaces simples, as mensagens Modbus são enviadas para a rede num *form* comum. Apesar da principal estrutura da mensagem do protocolo ser *peer-to-peer*, o Modbus é capaz de funcionar em ambas as redes *point-to-point* e *multi-drop*.

As ligações série Modbus podem ser feitas em dois modos básicos de transmissão, ASCII ou RTU (*Remote Terminal Unit*). O modo de transmissão em comunicações série define como as mensagens Modbus são codificadas. Usando Modbus/ASCII, a mensagem é codificada no formato ASCII, enquanto o formato Modbus/RTU usa código binário, o que reduz o seu tamanho permitindo maior transmissão de informação no mesmo espaço de tempo. Todos os nós do mesmo segmento de rede Modbus devem usar o mesmo modo de transmissão em série uma vez que um dispositivo configurado para funcionar em Modbus/ASCII não interpreta informação em Modbus/RTU, e vice-versa.

Em qualquer um dos modos de transmissão série (ASCII ou RTU), uma mensagem Modbus é colocada pelo transmissor num vetor que tem um início e fim conhecidos. Isto faz com que os recetores iniciem a leitura no início da mensagem, leiam o endereço e determinem qual o destinatário da mensagem, e saibam quando a mensagem terminou. Caso algo de errado aconteça durante a transmissão, a resposta resultará numa mensagem de erro.
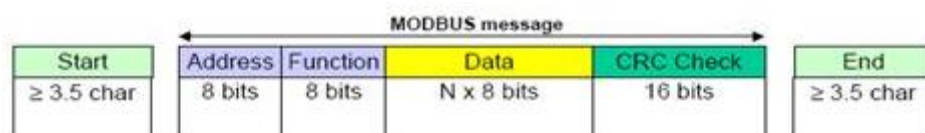
Figura 20 - Estrutura da Mensagem em Modbus RTU [28]

Start	Address	Function	Data	LRC	End
1 char :	2 chars	2 chars	0 up to 2x252 char(s)	2 chars	2 chars CR,LF
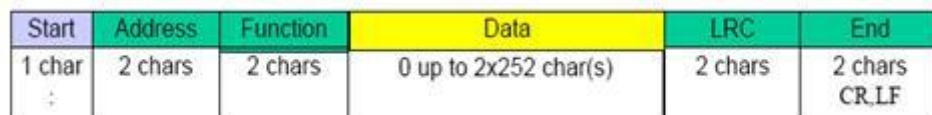
Figura 21 - Estrutura da Mensagem em Modbus ASCII [28]

Todas as mensagens Modbus têm a mesma estrutura, tendo quatro elementos básicos sempre presentes:

1. *Device Adress*
2. *Function Code*
3. *Data*
4. *Error Check*

A sequência destes elementos é sempre a mesma em todas as mensagens. O master da rede Modbus (pode haver mais do que um) inicia sempre a conversação, e aguarda a resposta do

escravo à qual a mensagem foi endereçada. Todos os outros escravos ignoram a mensagem se o endereço não lhes corresponder [25].

O primeiro parâmetro enviado em cada mensagem Modbus é o endereço do recetor. Este parâmetro contém um byte de informação. Os endereços válidos estão entre 0 e 247. Os valores entre 1 e 247 estão atribuídos para dispositivos Modbus individuais enquanto o valor 0 é utilizado para difundir mensagens para todos os escravos da rede [25]. Estes respondem sempre a uma mensagem, usando o mesmo endereço que o master aquando da pergunta. Desta forma, o master sabe sempre que dispositivo está a responder ao seu pedido.

O segundo parâmetro em cada mensagem Modbus é o código da função. Este define o tipo de ação a ser executada pelo escravo. Contém um byte de informação. Os códigos de função válidos vão desde 1 até 255 mas nem todos os dispositivos Modbus reconhecem o mesmo conjunto de códigos de função. A tabela 1 apresenta os mais comuns [27].

Tabela 1 - Códigos das funções Modbus [27]

Code	*Description*
01	*Read Coil Status*
02	*Read Input Status*
03	*Read Holding Registers*
04	*Read Input Registers*
05	*Force Single Coil*
06	*Preset Single Register*
07	*Read Exception Status*
08	*Loop detection (Diagnostic)*
15	*Force Multiple Coils*
16	*Preset Multiple Registers*
17	*Report Slave ID*

No protocolo Modbus existem duas formas de verificar se existem erros durante a transmissão ou não: paridade da mensagem e *frame checking*.

A verificação da paridade da mensagem pode ser adicionada opcionalmente. Existem dois tipos de código de paridade: a paridade par e a paridade ímpar. A paridade será par quando o número de bits de valor '1' for par; caso contrário, será ímpar. Por exemplo, na mensagem 10100101. Se utilizar paridade par, o bit de paridade terá o valor 0. Por outro lado, se a

paridade for impar, o bit de paridade terá o valor 1. Esta estratégia apenas deteta erros quando um número impar de bits foi alterado durante a transmissão, daí ser necessário a utilização de outros mecanismos [26].

Para além disso, *frame checking* é calculado e enviado no final da *frame*.

No caso de Modbus RTU, o método CRC (*Cyclical Redundancy Check*) é utilizado, enquanto em Modbus ASCII é utilizado o método LRC (*Longitudinal Redundancy Check*) [27]. Tanto o CRC como o LRC possuem 8 bits. O emissor da mensagem calcula os valores e envia-os no final da mensagem. O recetor recalcula o CRC/LRC e compara-o com o que foi enviado no final da mensagem. Se estes os dois valores não forem iguais, o recetor envia uma mensagem de erro [26].

2.10 TCP/IP

O TCP/IP é um conjunto de protocolos de comunicação entre computadores ligados numa rede (também chamado de pilha de protocolos TCP/IP). O nome vem de dois protocolos: o TCP (*Transmission Control Protocol*) e o IP (*Internet Protocol*).

Este protocolo funciona normalmente em conjunto com *Ethernet*, o seu suporte físico preferencial. Estes dois protocolos são o suporte para a Internet, utilizada abundantemente nos dias de hoje. Atualmente, grande parte dos sistemas operativos utilizados nos computadores já possui interfaces de TCP/IP instados em virtude da sua utilização pela Internet.

A figura 22 apresenta os *layers* inferiores do modelo OSI e os protocolos que podem ser utilizados para implementação destes.
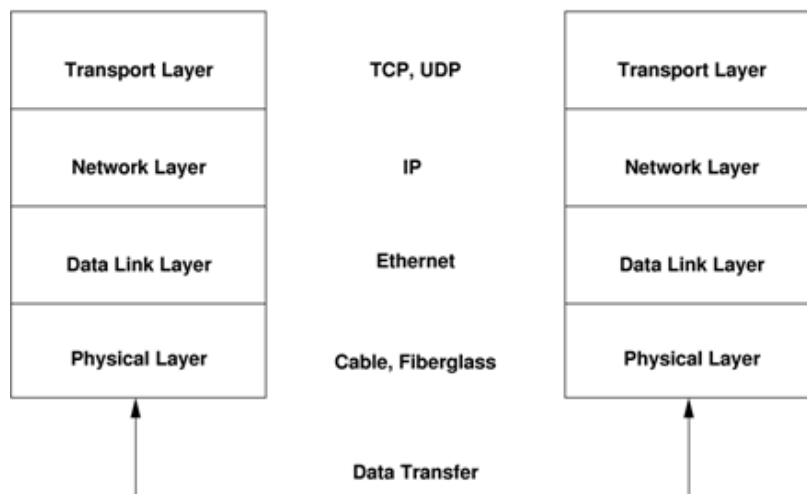
Figura 22 - Implementação comum do protocolo TCP/IP e *Ethernet*, com os respetivos layers do modelo OSI [31]

A figura 23, apresenta a estrutura do campo de dados de um *frame Ethernet.*

Figura 23- Campo de dados de um *frame Ethernet* (usando TCP/IP)

O protocolo TCP/IP pode ser dividido em dois protocolos, de forma a esquematizar o seu funcionamento:

- O protocolo TCP
- O protocolo IP

O IP (*Internet Protocol*) é um protocolo de nível da camada rede (*Network Layer*) do modelo OSI. A principal função do protocolo IP é transportar os *frames* de uma rede para outra. O protocolo IP fornece os seguintes serviços:

- Endereçamento: o cabeçalho IP possui o endereço IP do remetente e do destinatário da mensagem. Estes endereços são utilizados pelos *routers* intermédio para selecionar o caminho a seguir pelos dados.
- Fragmentação: na grande maioria dos casos, a mensagem a enviar não cabe num pacote apenas. Assim eles têm de ser fragmentados em pacotes mais pequenos. Isto irá

permitir enviar grandes volumes de dados através da rede. O protocolo IP do destinatário irá "remontar" os diferentes fragmentos da mensagem enviada.

- *Packet timeout*: cada pacote possui um campo TTL (Time To Live) que é decrementado sempre que passa por um *router*. Se o campo TTL atingir 0, o pacote é descartado evitando assim que ele viaje em círculos e obstrua o fluxo da rede.
- Tipo de Serviço: o protocolo IP suporta prioridades no tráfego, permitindo que alguns pacotes sejam rotulados de acordo com o serviço.
- Opções: o protocolo IP fornece muitas funções opcionais, como por exemplo especificação do caminho que o pacote deve seguir, ou verificação do caminho seguido pelo pacote, bem como algumas funções de segurança [30].

O endereçamento IP é um tema importante, já que é ele que permite que o grande número de redes que existem sejam capazes de comunicar entre si. Existem duas versões do protocolo IP: o IPV4, que é utilizado na grande maioria das situações, enquanto o IPV6 é a versão atualizada. No IPV4, os endereço IP são compostos por 4 blocos de 8 bits (32 bits no total), que são representados através de números de 0 a 255, como "200.156.23.43" ou "64.245.32.11". O IPV6 é uma expansão do anterior e foi desenvolvido para ultrapassar algumas das limitações do IPV4. A mais importante é o tamanho do endereço de IP, passando de 32-bits para 128 bits, uma vez que os endereços IPV4 começam a ser escassos.

A figura 24 apresenta os conteúdos enviados num cabeçalho IP [30].

O protocolo IP não possui mecanismos de retransmissão nem dá garantia de uma transmissão integra ou ordenada [32]. Assim, todas as questões de consistências para a integridade dos dados transmitidos terão de ser tratadas com o protocolo TCP.

Figura 24 - Conteúdos de um cabeçalho IP [32].

O TCP (*Transmission Control Protocol*, ou em português, Protocolo de Controlo da Transmissão) é um protocolo de nível da camada de transporte (camada 4) do Modelo OSI.
Durante a receção de dados, o protocolo TCP processa os pacotes enviados pela camada inferior do modelo OSI (normalmente, o protocolo IP), colocando-os pela ordem correta (já que os pacotes podem chegar ao destinatário fora de ordem), verifica se os dados dentro dos pacotes estão íntegros e envia um sinal de confirmação ao transmissor, confirmando que o pacote foi recebido corretamente e sem erros. Se nenhum sinal de confirmação (*acknowledge*) for recebido (ou porque o pacote não chegou ao destinatário ou porque os dados estavam corrompidos), o transmissor enviará novamente o pacote perdido [29].
Enquanto que o TCP reordena os pacotes e usa mecanismo de confirmação de receção – desejável na transmissão de dados – existe um outro protocolo que opera nesta camada que não tem esses recursos, o protocolo UDP (*User Datagram Protocol*). Por essa razão o TCP é considerado um protocolo confiável, enquanto que o UDP é considerado um protocolo não confiável. O UDP é tipicamente usado quando nenhum dado importante está sendo transmitido. Como o UDP não reordena os pacotes e nem usa mecanismo de confirmação, ele é mais rápido do que o TCP.

Durante a transmissão de dados, o TCP irá receber os dados passados da camada de Aplicação (*Application Layer*) e irá adicionar a estes um cabeçalho. Na receção de dados, o cabeçalho será removido antes de os dados serem enviados para a porta apropriada. Este cabeçalho possui várias informações de controlo, em particular o número da porta de origem, o número

da porta de destino, um número de sequência (para a confirmação de receção e mecanismos de reordenamento usado pelo TCP) e uma soma de verificação (CRC, cálculo utilizado para verificar se os dados foram recebidos corretamente no destino). O cabeçalho TCP tem entre 20 e 24 bytes (dependendo se o campo opções estiver sendo ou não usado) e a sua estrutura é apresentada na figura 22.
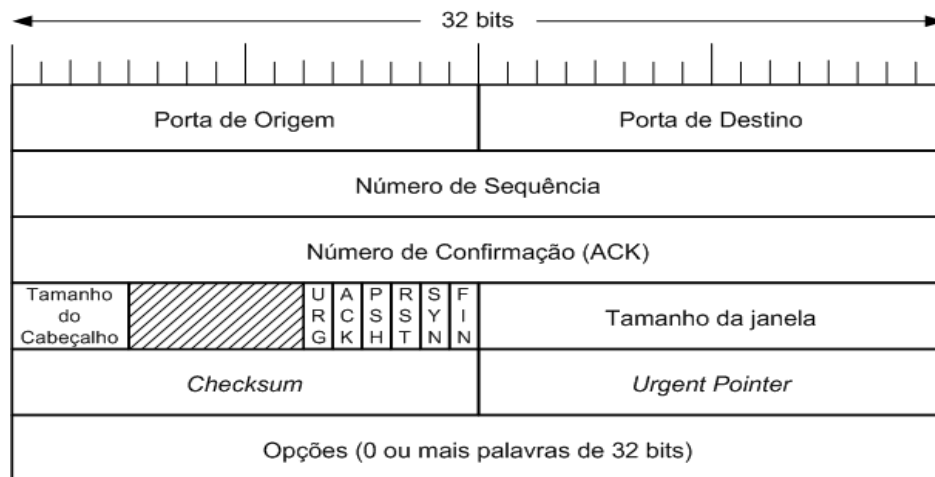
Figura 25 - Estrutura cabeçalho de TCP [34]

Assim sendo, a estrutura de uma mensagem utilizando os protocolos TCP, IP e *Ethernet* tem a estrutura apresentada na figura 23.
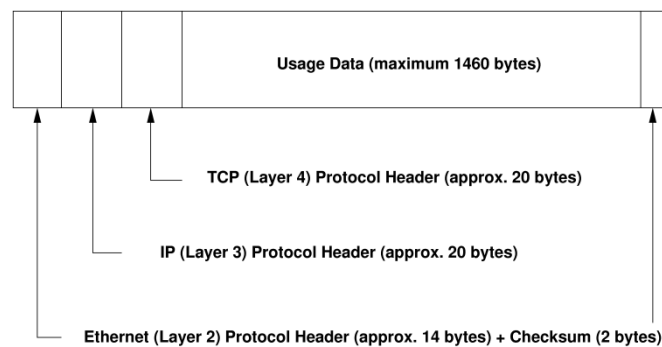
Figura 26 - Estrutura de uma mensagem usando TCP/IP e *Ethernet* [35]

Capítulo 3
Conceção e Especificação do sistema

3.1 Arquitetura do sistema de controlo

A arquitetura do sistema de controlo desenvolvida no âmbito desta dissertação é apresentada na figura 27.
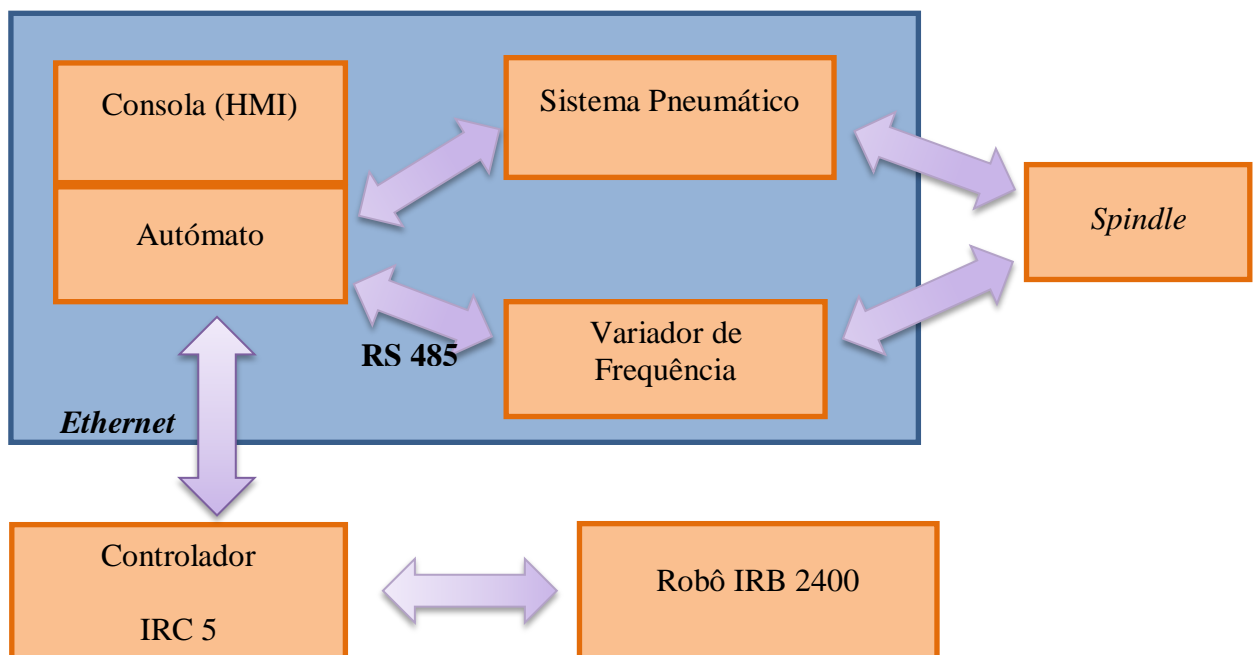
Figura 27 – Diagrama do Sistema de Controlo a Implementar

Este sistema está baseado num autómato com uma HMI integrada. O autómato comunica com o sistema pneumático (responsável pelo fornecimento de ar ao *spindle* para refrigeração e acionamento do sistema de troca de ferramenta) através de linhas de I/O dedicadas. Este autómato comunica também com o variador de frequência através de uma ligação física RS485 e com o controlador do robô por uma ligação *Ethernet*.

Uma alternativa a esta arquitetura de controlo poderia passar pela utilização do controlador do robô em substituição do autómato. Contudo esse processo seria mais dispendioso, uma vez que seria necessário adquirir cartas de expansão para o controlador do robô, seria necessário adquirir um conversor *Ethernet*/RS485, a configuração das comunicações e a programação das diferentes tarefas seriam mais complexas, ficando o sistema sempre dependente do controlador do robô.

3.2 Principais componentes utilizados

Para a implementação a arquitetura apresentada no subcapítulo anterior, serão usados os seguintes equipamentos:

3.2.1 Robô ABB IRB 2400

O robô utilizado é um ABB IRB 2400/16 (figura 28), com uma capacidade de carga de 20kg e um alcance de 1,5m. É um modelo recomendado para aplicações de soldadura a arco, operações de corte e de rebarbagem, manuseio e montagem de componentes [29]. Algumas características do robô são apresentadas na tabela 2.

Tabela 2 - Características do robô ABB IRB 2400/16 [29]

Alcance	1,55m
Capacidade de Carga	20 Kg
Número de Eixos	6
Peso	380 Kg
Repetibilidade de Posição	<0.07 mm
Repetibilidade de Trajetórias	<0.15 mm
Velocidade Máxima	450º/s no Eixo 6, 360º/s nos Eixos 4 e 5 e 150º/s nos Eixos 1,2 e 3.

Figura 28 - Robô ABB 2400

3.2.2 Controlador ABB IRC5

Este controlador é a quinta geração de controladores robóticos da ABB e sucede ao S4. Possui avanços significativos na precisão, previsão e repetibilidade de trajetórias e pontos relativamente ao seu antecessor. Suporta a linguagem de programação RAPID (desenvolvida pela ABB). Possui até 8192 entradas e saídas, 6 entradas analógicas, 1 porta de comunicação série RS-232, suporta redes de comunicação DeviceNet, Profinet e *Ethernet*/IP e possui também interfaces para controlo de força, sistemas de visão e *conveyor tracking*.

3.2.3 *Spindle* PDS XLC-070

O *spindle* utilizado é um equipamento da marca PDS (*Precise Drive Systems*), modelo XLC-070 com uma potência de 2,2 KW (3 HP). Este modelo é alimentado a uma tensão de 380V, corrente máxima admissível de 4,1A, velocidade máxima de 40000 rpm e binário máximo de 0,97 Nm. Este *spindle* possui um sistema automático de mudança de ferramenta (HSK-E32) acionado pneumaticamente. O arrefecimento do *spindle* é feito por ar comprimido.
É apresentada na figura 29, uma imagem deste equipamento bem como a legenda dos seus principais componentes.
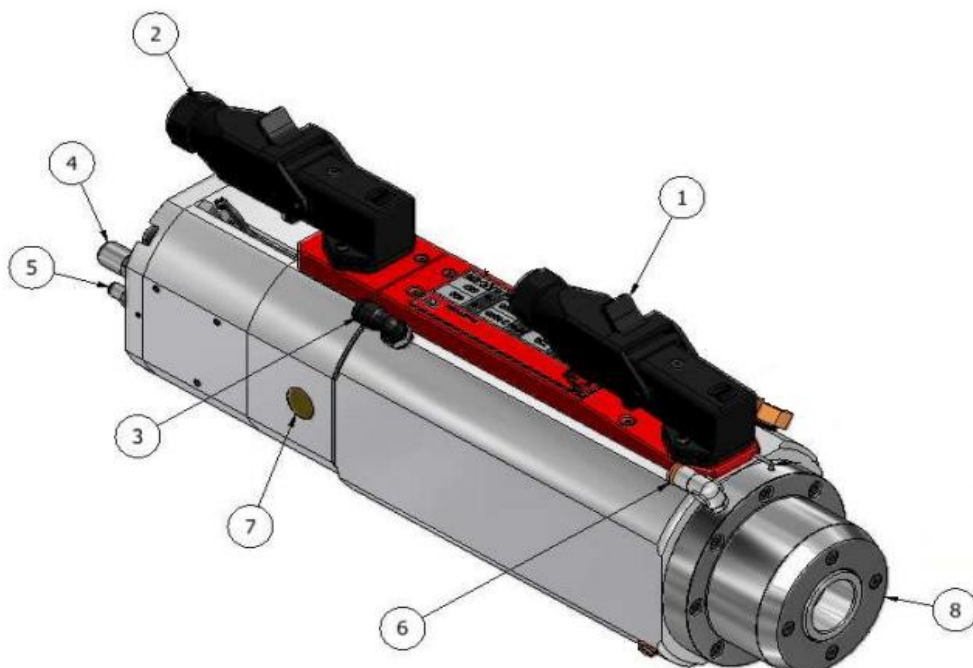
Figura 29 – *Spindle* PDS XLC 070 [36]

Tabela 3 - Elementos importantes presentes no *Spindle* XLC 070 [36]

1	Ficha de Potência	Conector com 5 pinos para 3 fases elétricas e sensor de temperatura.
2	Ficha de Comando	Conector com 6 pinos para alimentação e monitorização dos sensores existentes.
3	Conector Ar de 6mm	Ar a 6 bar, caudal de 155l/min para arrefecimento e pressurização.
4	Conector Ar de 6mm	Ar seco e limpo a 6 bar para atuação da ferramenta.
5	Conector Ar de 4mm	Ar seco e limpo a 2 bar para limpeza.
6	Conector Ar de 4mm	Ar seco e limpo a 2 bar para pressurização.
7	Escape	Escape para ar de arrefecimento e atuação da ferramenta.
8	Suporte Rolamento Frontal	Placa exterior de suporte.

Este *spindle* é recomendado para a maquinagem de diversos tipos de madeiras (contraplacado, MDF, aglomerados de madeira, pinho, madeiras macias) e alguns tipos de polímeros (PVC, plásticos flexíveis, espumas rígidas). É também possível a maquinagem de madeiras rígidas (carvalho, madeiras duras, macieira), ligas leves, alumínio e plásticos duros mas necessita de especial cuidado durante a maquinagem. Não é adequado para ligas de titânio e níquel, aços, granito e pedra [36].

Este *spindle* possui 4 sensores PNP que permitem monitorizar o seu funcionamento e o sistema automático de mudança de ferramenta.

O sensor S1 regista a rotação do *spindle*. Disponibiliza 2 impulsos por rotação do eixo do *spindle*. Quando este sensor regista rotação, não é seguro proceder a mudança da ferramenta a ser utilizada.

Figura 30 - Esquema do sistema de mudança de ferramenta [36]

O propósito do sensor S2 é registar se o sistema de fixação da ferramenta está aberto ou fechado. É obrigatório que o sistema de fixação da ferramenta esteja corretamente fechado para uma utilização segura do equipamento.

O sensor S3 regista se a ferramenta está corretamente

fixa. O sensor não irá ser atuado se existir uma má fixação devido a detritos no cone ou outra qualquer interferência.

O sensor S4 verifica se o êmbolo do atuador de fixação da ferramenta se encontra completamente recuado. Isto assegura que o êmbolo de atuação e o veio de rotação do *spindle* não estão em contacto. O contacto entre estes dois componentes pode levar a soldadura por fricção, o que pode danificar ambos os componentes como esquematizado na figura 30.

A tabela 4 apresenta as possíveis combinações dos sinais dos 3 sensores (S2, S3 e S4) necessários para controlo correto do sistema automático de mudança de ferramenta.

Tabela 4 - Possíveis combinações dos sinais sensores e condições correspondentes

Condição	S2	S3	S4
Sistema Fixação Aberto	ON	OFF	OFF
Sistema Fixação Fechado sem Ferramenta	OFF	OFF	ON
Ferramenta Presa, Embolo não Recuado	OFF	ON	OFF
Ferramenta Presa, Embolo Recuado	OFF	ON	ON

O modelo existente no laboratório não possui o sensor RTD de temperatura apresentado no manual [36]. Assim, a monitorização detalhada da temperatura no interior do *spindle* não é possível. Em vez disso, para a monitorização da temperatura do *spindle* será usado um contacto térmico localizado no interior do estator do motor AC, normalmente fechado e projetado para abrir a 130ºC. Quando este contacto abrir, a utilização do *spindle* não será permitida.

Antes do início das operações com o equipamento deverá ser realizado um ciclo de aquecimento do *spindle*, que dependerá do tempo de paragem do *spindle*. A tabela 5 indica a sequência de ações que deve ser realizada de acordo com o ciclo a efetuar.

Tabela 5 - Ciclos de Aquecimento do *Spindle* [36]

	Ciclo de Aquecimento Normal	Aquecimento Prolongado (paragem superior a 6 meses)
1º	5 Minutos a 20000 rpm	25 minutos a 8000 rpm
2º	3 Minutos a 30000 rpm	10 minutos a 20000 rpm
3º	3 Minutos a 40000 rpm	10 minutos a 30000 rpm
4º		15 minutos a 40000 rpm

Mais informações sobre a manutenção, aquecimento e operação do *spindle*, bem como informações mais detalhadas sobre o equipamento, podem ser encontradas em [36].

3.2.4 Autómato Unitronics Vision 350

Após uma análise dos autómatos existentes no mercado, nomeadamente os das marcas Omron, Schneider, Unitronics e Siemens, a escolha recaiu no autómato da marca Unitronics modelo V350. Este modelo possui todas as funcionalidades de interfaces de comunicação necessárias para a aplicação e possui uma HMI integrada o que permite ter uma relação custo/funcionalidade vantajosa.

O autómato selecionado, Unitronics Vision 350-35-R34 (figura 31), possui 20 entradas digitais, 2 entradas Analógicas/Digitais, 12 Saídas por relés e possui uma porta RS485. A porta *Ethernet* é adquirida a parte. Possui memória com capacidade para 8192 variáveis, 512 *long integers,* 256 *double integers* e 384 contadores. Este modelo suporta *Ethernet* via TCP/IP, protocolo MODBUS e CANbus. Possui a opção de Web server, envio automático de emails e SMS. [37].

Quanto à HMI integrada, é TFT LCD, possui 65536 cores, tamanho de 3,5” e uma resolução de 320*240 *pixels*. Existem ainda 5 botões programáveis que podem ser utilizados para navegação ou associados a botões e variáveis.

Figura 31 - Autómato Unitronics Vision 350

3.2.5 Variador de Frequência

O variador de frequência utilizado é um variador Delta VFD-VE (figura 32). Este modelo foi recomendado pelo fornecedor do *Spindle* e foi adquirido ao mesmo tempo que este.
O variador Delta VFD-VE (VFD037V43A-2, *Frame* B) é adequado para motores de 0.75 KW a 3.7 KW.
Este variador possui as seguintes características [32, 33] (ver tabela 6):

- Economia automática de energia a carga leve.
- Proteção contra o sobreaquecimento do motor (função PTC).
- Função de desaceleração de energia de travagem.
- Controlo de posição.
- Contro de malha fechada, faixa de controlo de velocidade 1:1000.
- Controlo PID integrado.
- 4 Ajustes de tempo de curva-S/aceleração/desaceleração independentes.
- Sintonização automática de parâmetros do motor e inércia da carga.
- Controlo de posição por impulso E/S.

Figura 32 - Variador de Frequência Delta VFD-VE [38]

Tabela 6 - Características Variador Delta VFD-VE VFD037V43A-2 [40]

Classe de Tensão		Classe 460V
Classificação de Saída	Capacidade de Saída Nominal	6.3 kVA
	Corrente de Saída Nominal para Binário Constante	8.5 A
	Saída Máxima Aplicável ao Motor	5.0 HP
	Corrente de Saída Nominal para Binário Variável	10 A
	Saída máxima Aplicável ao Motor	7.5 HP
	Tensão de Saída Máxima	Trifásico Proporcional à Tensão de Entrada
	Frequência de Saída	0.00 ~ 600.00 Hz
Classificação de Entrada	Corrente de Entrada Nominal	9.9 A
	Tensão/Frequência Nominal	Trifásico, 380 ~ 480V
	Tolerância de Voltagem	+/- 10%
	Tolerância de Frequência	+/- 5%
Método de Arrefecimento		Ventilador
Peso		6.8 KG

Possui também uma porta RS485, a qual será utilizada para comunicação entre o variador de frequência e o autómato anteriormente apresentado.

3.3 Especificação dos requisitos do sistema

Tendo por base as especificações genéricas da aplicação de controlo a desenvolver (apresentada no subcapítulo 1.2 – Objetivos) foram elaboradas as seguintes especificações funcionais que serviram de base ao desenvolvimento da programação do autómato, à conceção dos circuitos elétricos do armário de controlo e à configuração do variador de frequência.

- O equipamento pode operar num modo "Manual" e num modo "Automático".
    - No modo "Automático", o controlador do robô deverá enviar os parâmetros de funcionamento necessários (velocidade, sentido de rotação, etc) para o autómato, não sendo possível a alteração de parâmetros na HMI.
    - No modo manual, os parâmetros de funcionamento do *spindle* podem ser alterados a qualquer altura na HMI.
- O sistema é iniciado quando o interruptor seccionador é movido para a posição ON.
- Os parâmetros de funcionamento que podem ser controlados são: a velocidade de rotação, o sentido de rotação, abertura e fecho do sistema para mudança de ferramenta, rampa de aceleração e rampa de desaceleração.
- O autómato controla a operação ligar e desligar o variador de frequência.
- A atuação de qualquer um dos botões de emergência provoca a paragem da movimentação do robô e da rotação do *spindle*. Para voltar a utilizar os equipamentos, é necessário colocar os botões de emergência não atuados e, após, confirmar a reativação num botão existente na HMI.
- Quando existe algum problema este pode ser identificado pelo sistema (como por exemplo, temperatura excessiva no spindle, ar comprimido à pressão inferior a recomendada, botão de emergência atuado, etc). A descrição do problema é identificada na HMI.
- Antes de poder ser utilizado o *spindle* em pleno funcionamento, o autómato executa o ciclo de aquecimento do *spindle*. Após terminado este ciclo, é indicado na HMI que o *spindle* está pronto a ser utilizado.
- O sinalizador luminoso existente no armário elétrico, está ligado quando o *spindle* está em rotação.

- As condições seguintes impedem o arranque do *spindle* ou provocam a sua paragem caso já esteja em funcionamento:
    - Não se encontrar uma ferramenta montada ou o sistema de mudança de ferramenta se encontrar aberto.
    - A temperatura no seu interior for superior a 130ºC.
    - Não existir ar comprimido às pressões recomendadas.
    - Algum dos botões de emergência estar atuado.
    - Algum equipamento não estiver a comunicar devidamente.
    - A combinação dos estados dos sensores existentes no *spindle* não for a correta.
    - Outra qualquer condição que não garanta a segurança na sua utilização.

Capítulo 4
Implementação do sistema

4.1 Softwares utilizados

Para a implementação do sistema de controlo foram usados quatro *softwares* que se revelaram importantes para a programação e resolução de problemas que surgiram durante o desenvolvimento do sistema. Estes softwares foram:

- VisiLogic;
- RobotStudio;
- VFD Soft;
- Windmill ComDebug.

O VisiLogic é um *software* da Unitronics, desenvolvido para a programação dos autómatos da referida marca. Este *software* possui os elementos de programação normais dentro deste tipo de softwares como contactos abertos ou fechados, blocos para cálculo matemático (somas, multiplicações, etc) e blocos de conversão de variáveis, etc, e possui também blocos para configuração das comunicações a efetuar que se revelaram bastante uteis durante a fase de programação. É um *software* bastante intuitivo e fácil de utilizar.
Foi utilizado para programar todas etapas e ações dos GRAFCETs implementados, bem como para construir todos os ecrãs que surgem na HMI e implementar todas as comunicações com o variador de frequência e o controlador do robô.

O *software* RobotStudio é um ambiente virtual de simulação e programação offline desenvolvido pela ABB. Permite que o programador crie cenários com modelos de unidades ABB (tanto a nível de representação 3D como de emulação física). Este *software* é construído com base na tecnologia VirtualController ABB, uma cópia exata do software real que

comanda os seus robôs no processo de produção. Deste modo, possibilita simulações muito realistas, utilizando programas de robôs verdadeiros, idênticos aos utilizados numa fábrica. A linguagem de programação de robôs suportada é o RAPID. A figura 33 apresenta uma imagem do ambiente de trabalho deste *software*. Este *software* foi utilizado para programar rotinas (em linguagem RAPID) que permitem controlar a operação do spindle, quando em funcionamento automático.
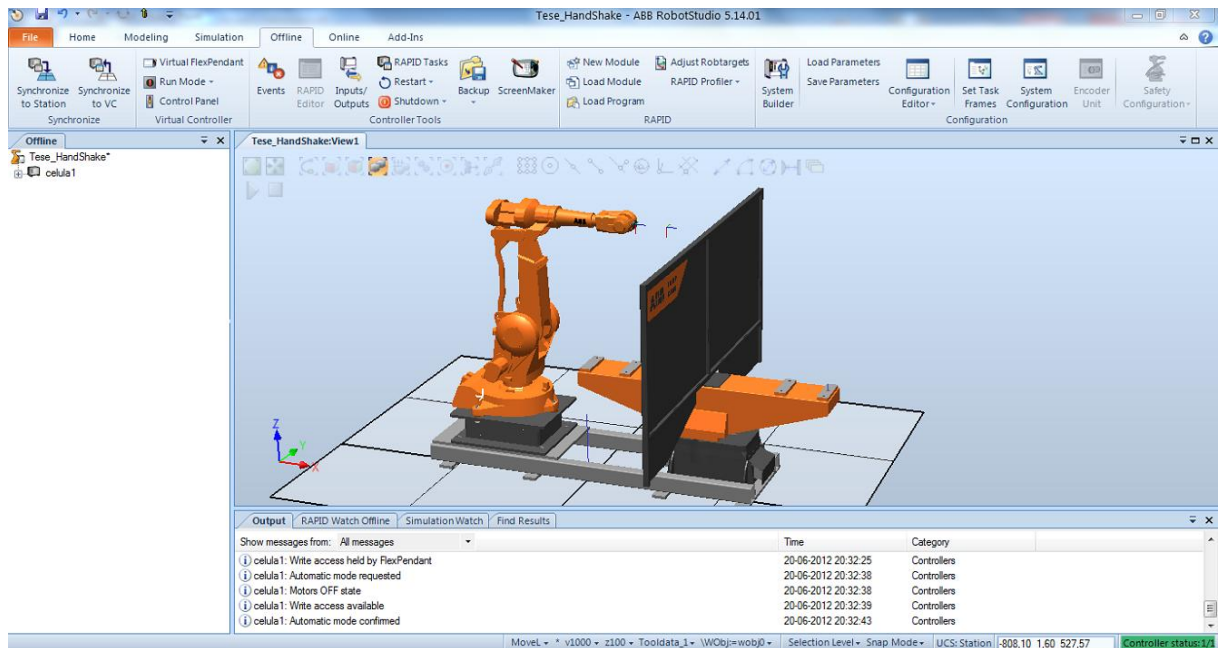
Figura 33 - Ambiente de Trabalho do *software* RobotStudio

O software DeltaSoft é um software disponibilizado pela Delta Electronics, inc que pode ser utilizado nos variadores de frequência da família Delta VFD-B, VFD-F, VFD-M, VFD-S, VFD-E, VFD-L e VFD-VE. Este *software* permite fazer *download* de todos os parâmetros de configuração do variador de frequência, que podem ser alterados de forma mais fácil e comoda e posteriormente feito o *upload* dos novos parâmetros. Este *software* possui também uma melhor explicação das funções dos diversos parâmetros. Uma vez que estes variadores de frequência possuem uma porta RS485 e os computadores normalmente possuem portas RS232, foi também necessária a utilização de um conversor RS232-RS485 para a sua ligação. Este programa tem funcionalidades que permitem operar o variador podendo ser dadas ordens simples como arrancar, parar, inverter sentido, o que se revelou muito útil na verificação dos dados a enviar em cada mensagem. Estas funcionalidades foram também uteis para teste de mensagens para configuração de parâmetros como potencia do motor, frequência máxima, corrente e tensão máximas, entre outros.

O Windmill ComDebug é um *software* gratuito que permite comunicar com praticamente qualquer dispositivo que suporte RS232, RS422, RS485 ou Modbus. É possível introduzir e exibir os dados em formato hexadecimal, ASCII ou binário. Com este *software* é possível enviar mensagens para um dispositivo que esteja conectado à porta série do computador e ver qual a resposta que o dispositivo envia. Foi muito útil na implementação da comunicação autómato-variador de frequência. Uma imagem deste software é apresentada na figura 34.
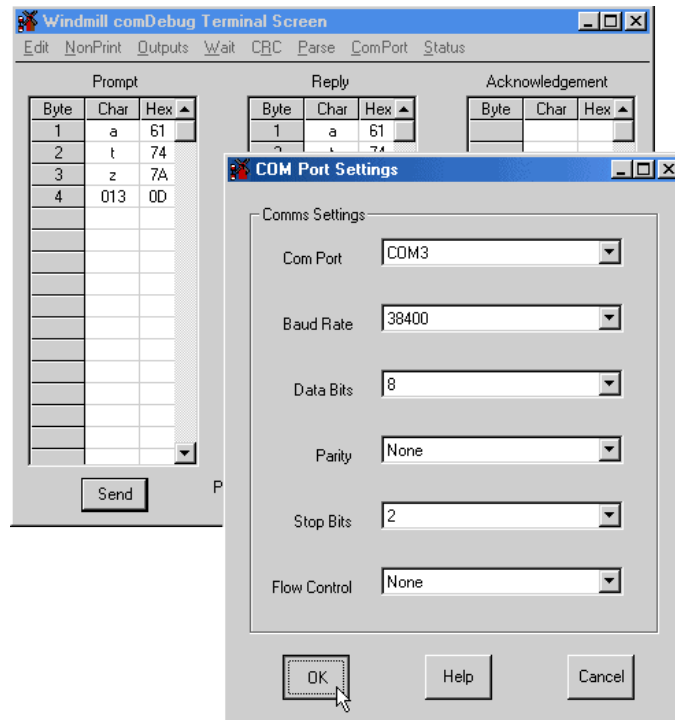
Figura 34 - Ambiente do software Windmill ComDebug

4.2 Instalação elétrica e pneumática

O autómato e o variador de frequência, juntamente com outros dispositivos de comando e proteção, foram instalados num quadro elétrico. Foi também concebido e projetado um esquema elétrico (apresentado no anexo A) e que serviu de base para a montagem do quadro elétrico. Neste esquema elétrico foram utilizados alguns componentes elétricos, que por serem muito comuns, não serão abordados aqui com grande detalhe.

Os componentes presentes no esquema elétrico e instalados no quadro elétrico foram:

Comando e Proteção:

- ✓ Interruptor seccionador de 4 polos
- ✓ Porta fusíveis para fusíveis de dimensão 10mm*38mm
- ✓ 4 Fusíveis de cartuxo (dimensão 10mm*38mm) de 4 A, Cooper Bussmann
- ✓ Interruptor diferencial Kopp 25A, sensibilidade ao disparo de 20 mA
- ✓ Contator 3P com 1 contacto auxiliar, marca Telemecanique
- ✓ Disjuntor 5A
- ✓ Fusível de cartuxo (5mm*20mm) 2A
- ✓ Botão Emergência com 3 contactos normalmente fechados, marca Telemecanique

Variador de Frequência e Autómato:

- ✓ Variador de frequência Delta VFD-VE VFD037V43A-2
- ✓ Autómato Unitronics Vision 350

Fonte de alimentação:

- ✓ Fonte SIEMENS SITOP PSU100C 24 V/ 2,5 A

Conectores elétricos e sinalizadores luminosos:

- ✓ Conector fêmea de 9 pinos
- ✓ Conector fêmea de 25 pinos, Harting
- ✓ Sinalizador Luminoso de cor verde, marca Telemecanique

Foi instalado um botão de emergência que está ligado ao controlador ABB IRC5 de modo a que, quando atuado, tanto o robô como o *spindle* parem o funcionamento.

Os conectores que foram instalados para o exterior são também de tipo fêmea, de modo a assegurar que nenhum utilizador apanhe um choque elétrico por contacto com os pinos dos conectores.

O sinalizador luminoso instalado indica ao utilizador que o *spindle* se encontra em funcionamento.

A figura 35 apresenta o aspeto final do interior e exterior do quadro elétrico.

Figura 35- Aspeto final dos componentes montados no interior (à esquerda) e exterior (à direita) do quadro elétrico

Sistema Pneumático

A qualidade do ar utilizado no *spindle* é fundamental para o seu correto funcionamento e longevidade. Em equipamentos que utilizam ar comprimido, a probabilidade ocorrer condensação da humidade presente no ar devido a variações de pressão e temperatura são significativas. Isto pode prejudicar gravemente o funcionamento dos rolamentos de precisão do *spindle* e a fiabilidade do sistema de fixação da ferramenta.

Assim, é recomendado pelo fabricante que o ar utilizado cumpra a norma ISO 8573.1 Classe 1, como apresentado na tabela 7.

O laboratório onde o *spindle* está instalado já possui uma rede de ar comprimido que fornece ar a uma pressão de aproximadamente 7 bar. Como referido anteriormente, o *spindle* necessita de ar comprimido a 6 e a 2 bar.

Tabela 7 - Características necessárias no ar comprimido a ser utilizado no *spindle*

Dimensão aceitável das partículas	0.1 mm
Concentração aceitável de partículas	0.1 mg/m^3
Dimensão aceitável das partículas de lubrificante	0.01 mg/m^3
Ponto de condensação a 6 bar	3ºC
Ponto de condensação a pressão atmosférica	-22ºC

Com vista a cumprir os requisitos da norma ISO 8573.1 Classe 1 e os requisitos do *spindle* foi projetado um circuito pneumático (figura 36), que permite tratar e controlar o fluxo de ar a fornecer ao *spindle* durante o seu funcionamento.
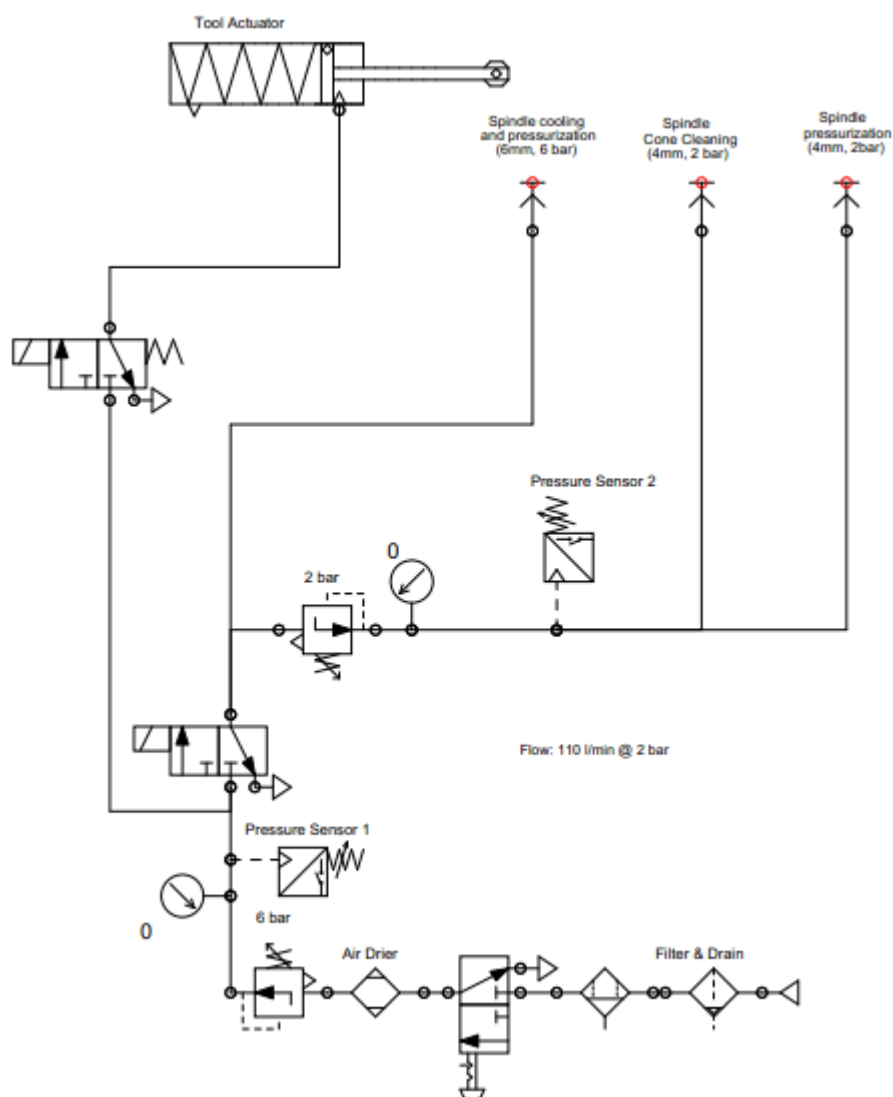
Figura 36 - Circuito Pneumático Utilizado

Este sistema incorpora numa unidade de tratamento de ar e num conjunto de electroválvulas e pressostatos que levarão o fluxo de ar até ao *spindle*. A unidade de tratamento de ar está instalada fora do armário elétrico, e é constituída por um um filtro de partículas de dimensões superior a 1 µm, um redutor de pressão (ajustado para 6 bar), um filtro de coalescência, uma válvula de fecho manual, e um filtro de secagem (*drier*).

A figura 35 mostra a unidade de tratamento de ar, enquanto a figura 36 mostra os restantes componentes.

Figura 37 - Unidade de tratamento de ar utilizada
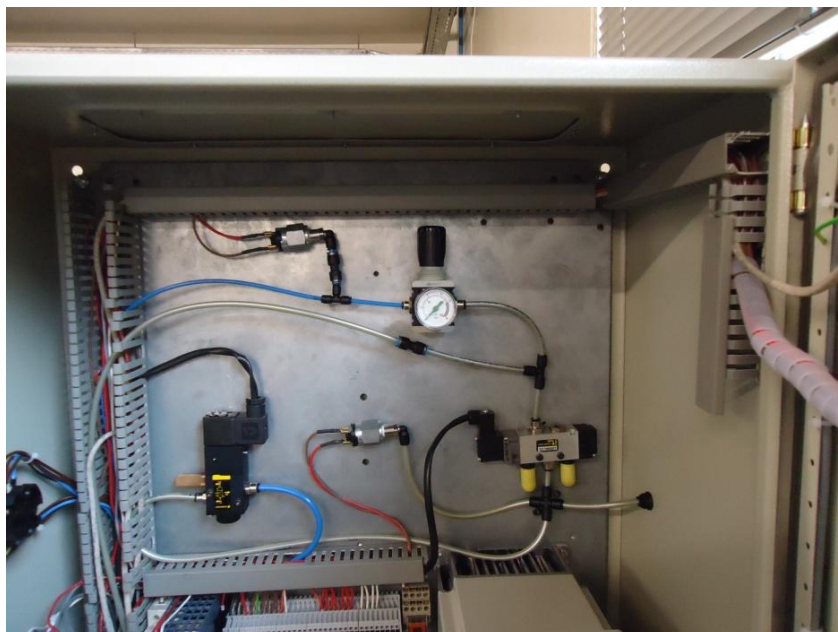
Figura 38 - Sistema pneumático

4.3 Programação do Autómato

4.3.1 GRAFCET Implementado

O Grafcet é uma metodologia que surgiu com a necessidade do desenvolvimento de programas para controlo de processos sequenciais. Ele não pretende minimizar as funções lógicas que representam a dinâmica do sistema. O seu potencial reside na imposição de um funcionamento rigoroso, evitando desta forma incoerências, bloqueios ou conflitos durante o funcionamento do mesmo. Para além disso, é uma metodologia de programação estruturada, que permite uma apresentação sintética do sistema de forma bastante clara. Por estes motivos, foi adotada esta metodologia para implementação dos requisitos e funcionalidades propostas no capítulo anterior.

O autómato foi programado com programação assíncrona de modo a permitir mais do que uma transição de estado por cada ciclo de funcionamento do autómato. Este facto é importante no funcionamento do GRAFCET apresentado na figura 39. O GRAFCET da figura 40 apresenta o funcionamento que o sistema terá quando se encontra no “Modo Normal”.

Neste tipo de sistemas que possui ferramentas de corte em rotação a elevadas velocidades, a segurança do sistema e dos utilizadores do equipamento é um fator essencial e indispensável de ser assegurado. Por este motivo foram desenvolvidos dois GRAFCETs que funcionam em paralelo, sendo que um deles trata da segurança do sistema (figura 39) e o outro da sequência correta de funcionamento (figura 40). O GRAFCET 1 implementa os dois modos de funcionamento: o modo normal e o modo de segurança. O GRAFCET 2 implementa a sequência dos ciclos de aquecimento e o funcionamento manual e automático.
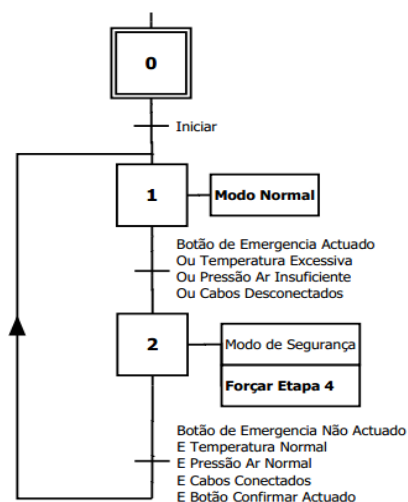
Figura 39 - GRAFCET 1: modos de funcionamento
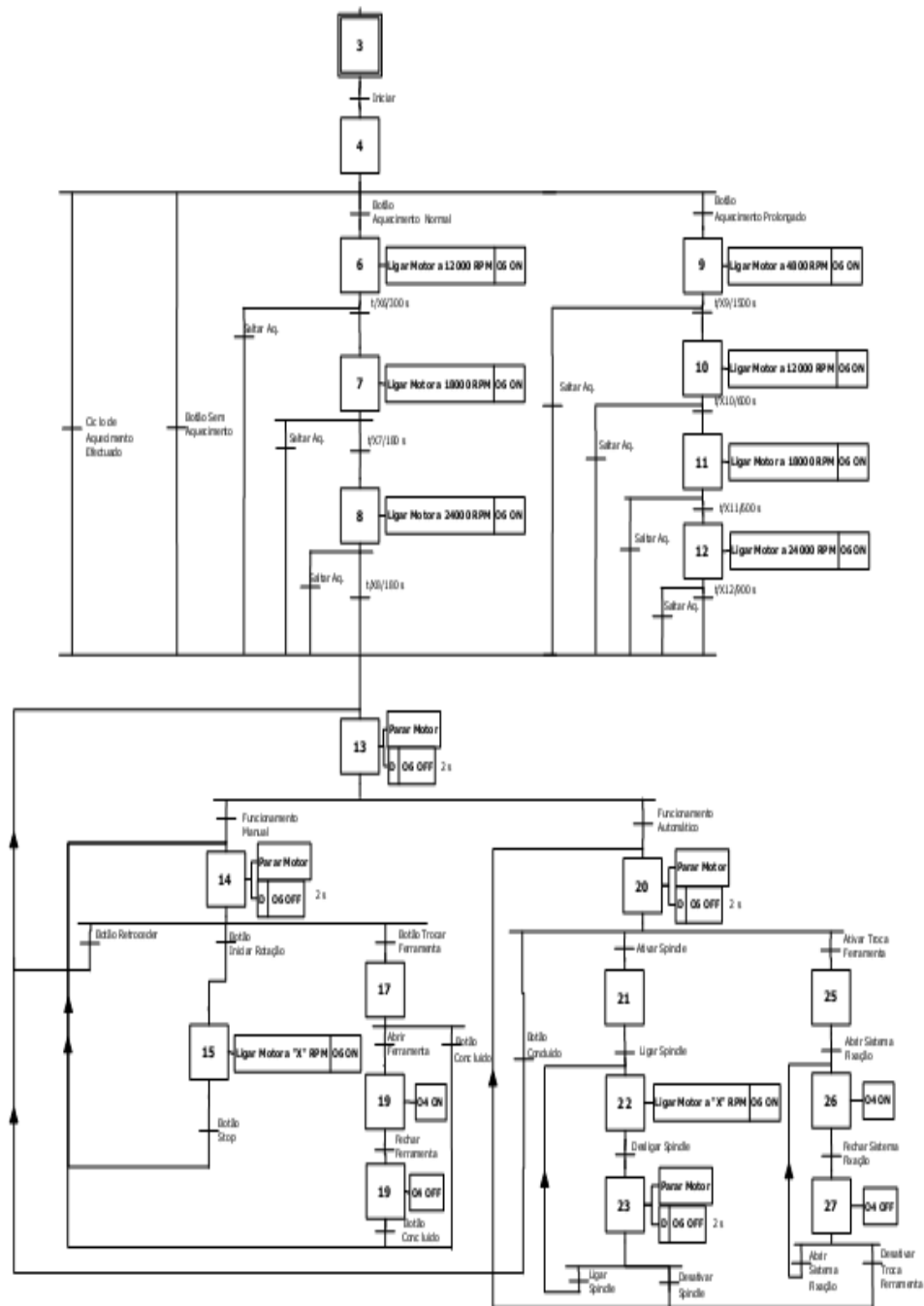
Figura 40 - GRAFCET 2: sequncia de operação

4.3.2 Implementação dos Modos de Funcionamento do sistema

O funcionamento do sistema foi estruturado a partir da criação de dois modos de funcionamento: Modo Normal e Modo de Segurança, que se encontram representados no GRAFCET da figura 39.

Os modos de funcionamento do sistema são controlados pelo GRAFCET 1 (figura 39).

O sistema é iniciado na etapa 0. Quando a condição de transição da etapa 0 é verdadeira, o sistema transita de etapa.

As condições monitorizadas para a segurança do sistema são:

- Estado do botão de segurança
- Estado do contacto que monitoriza a temperatura do *spindle*
- Estado de ligação dos cabos do armário elétrico ao *spindle*
- Existência de pressão no sistema pneumático instalado

Se alguma das condições indispensáveis à segurança do sistema não for verificada o sistema transita para a etapa 2, “Modo de Segurança”. Se todas as condições de segurança forem verificadas, o sistema transita para a etapa 1, que foi chamado de “Modo Normal”. Isto acontece porque foi utilizada programação assíncrona na programação do PLC. Se fosse utilizada programação síncrona que permite apenas uma transição por ciclo de funcionamento, o sistema iria percorrer um ciclo de funcionamento em “Modo Normal” mesmo que uma das condições de segurança fosse falsa, o que não cumpre com as especificações de segurança deste sistema.
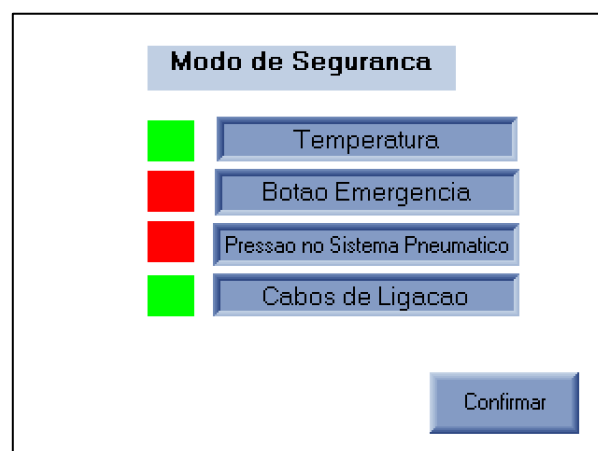
Figura 41- Ecrã apresentado no "Modo de Segurança"

Quando o sistema se encontra no "Modo de Segurança", o ecrã a figura 41 é apresentado na HMI. As condições de segurança não verificadas são indicadas com o indicador de cor vermelha, enquanto as verificadas são indicadas com a cor verde. O botão Confirmar estará visível apenas quando todas as condições de segurança estão verificadas. A atuação deste botão faz com que o sistema entre no "Modo Normal".

Uma vez no "Modo Normal" é possível operar normalmente o sistema de acordo com o apresentado no GRAFCET 2. Quando uma das condições de segurança não é verificada, o sistema entra no "Modo de Segurança", a movimentação do robô e a rotação do *spindle* são imediatamente paradas e o estado do GRAFCET 2 é forçado para a etapa 4 (etapa segura em que o sistema está parado) de modo a garantir o funcionamento seguro do sistema.

4.3.3 Implementação da comunicação Autómato - Variador de Frequência

O funcionamento do sistema envolve obviamente a troca de mensagens entre o variador de frequência e o autómato. A comunicação entre estes dispositivos é feita por RS485 e utilizando o protocolo MODBUS.

A porta RS485 do autómato e a porta RS485 do variador de frequência não apresentam o mesmo *pin out*. Assim, foi necessário fazer um cabo específico para esta aplicação, que está esquematizado na figura 42.

Foram utilizados os 4 terminais da porta do autómato dedicados a RS232 para serem ligados ao computador pela porta série existente para transferência dos programas, monitorização das variáveis do autómato e *debug*. Os 2 terminais restantes da porta do autómato são utilizados para a comunicação RS485 com o variador de frequência.
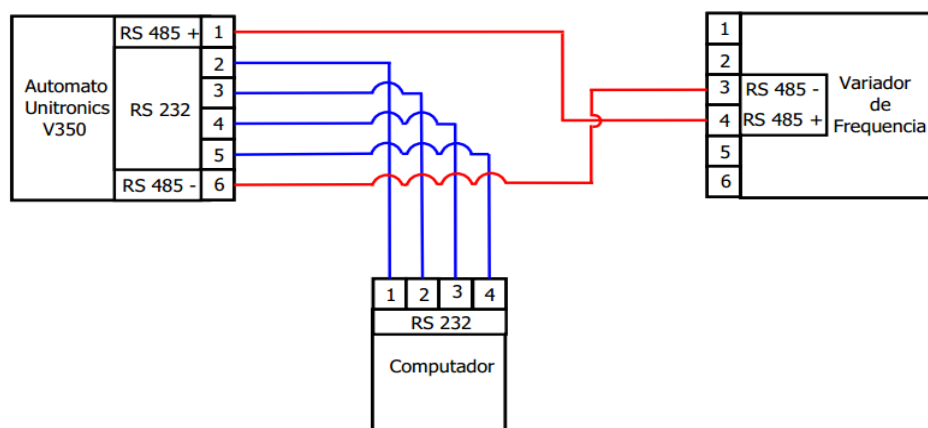
Figura 42 - Esquema do cabo utilizado nesta aplicação

A utilização deste cabo envolve a necessidade de alguns cuidados especiais, uma vez que não é possível a utilização simultânea de RS232 e RS485. Para além disso, estando o autómato configurado para utilizar a comunicação RS485, quando é utilizada a porta RS232 (por exemplo, o envio de um novo programa para o autómato) a comunicação RS485 fica automaticamente desativada. Para a comunicação RS485 ficar ativa novamente, é necessário reiniciar o autómato.

A monitorização dos parâmetros do autómato não é possível quando o equipamento está em funcionamento normal uma vez que não é possível a utilização simultânea de RS232 e RS485.

O protocolo Modbus é utilizado na troca de mensagens por RS485 entre o autómato e o variador de frequencia. Neste caso e uma vez que o autómato só suporta Modbus RTU, é utilizado este modo. São utilizados 8 bits na comunicação, paridade *none* e 1 *stop bit*. O sistema funciona segundo uma arquitetura *master/slave*, na qual o *master* é o autómato e o *slave* o variador de frequência. Quando o autómato envia uma mensagem para o variador de frequência, este último envia uma nova mensagem de volta a indicar se a mensagem foi recebida corretamente ou se ocorreu um erro.

Durante o funcionamento do sistema, diversas mensagens são trocadas entre estes equipamentos. A tabela 8, apresentada na página seguinte, exibe as mensagens mais comuns.

O endereço do variador de frequência é sempre 01, pelo que os primeiros dois dígitos nas mensagens trocadas tomem o valor 01. As unidades das incógnitas IIII, TTTT e CCCC na tabela 8 correspondem a 0,1 Hz, 0.1V e 0.01A respetivamente.

Os códigos de erro apresentados têm significados diferentes:

01 – Código de função inválido

02 – Endereço de dados inválidos

03 – Valor atribuídos aos dados inválidos

Para informações mais detalhadas sobre a comunicação entre estes equipamentos, deve consultar-se a referência [40].

Tabela 8 - Algumas das mensagens enviadas e recebidas pelo autómato durante o funcionamento do sistema (excluindo o LRC e End bits)

Tarefa	**Mensagem Enviada**	**Mensagem Recebida**
Alterar Frequência de Saída (IIII – frequência desejada)	01062001IIII	01062001IIII
Arranque	010620000002	010620000002
Paragem	010620000001	010620000001
Alterar Sentido	010620000040	010620000040
Ler Tensão de Saída (IIII- Valor de Tensão Recebido)	010321060001	010304TTTT
Ler Corrente Consumida (IIII- Valor de Corrente Recebido)	010321040001	010304CCCC
Mensagens de Erro (Enviadas Quando o Variador de Frequência Não Entende o Pedido, XX depende do Pedido Efetuado)		01XX01 01XX02 01XX03

4.3.4 Implementação dos Ciclos de Aquecimento do *spindle*

É indicado no manual de utilização do *spindle* a necessidade de proceder a um ciclo de aquecimento antes do equipamento ser utilizado. Se esta indicação não for seguida, o equipamento pode ser danificado e a garantia do fabricante deixa de ser válida.

Assim, quando o sistema é iniciado e a etapa 4 fica ativa, o ecrã mostrado na figura 43 é apresentado.
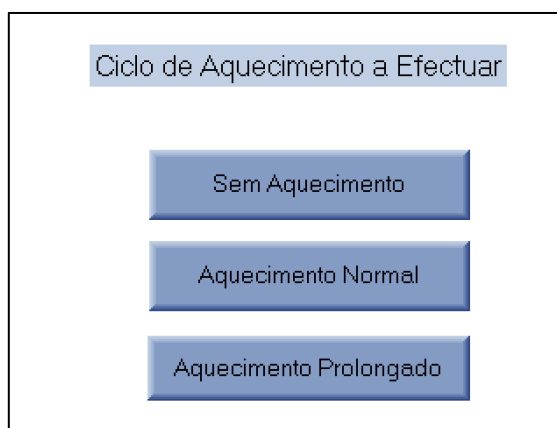
Figura 43- Ecrã apresentado para seleção do ciclo de aquecimento

O utilizador deverá indicar o ciclo de aquecimento a efetuar. O tempo e as velocidades que devem ser utilizadas em cada ciclo são descritos na tabela 5, apresentada no capítulo anterior.

Após a seleção do ciclo de aquecimento a efetuar, aparece um novo ecrã que indica quais as etapas que irão ser realizadas e sinaliza com um visto as que já foram concluídas. A qualquer altura é possível terminar o ciclo de aquecimento premindo o botão Saltar, como apresentado na figura 44.

Quando o ciclo de aquecimento é terminado, o sistema encontra-se pronto a ser utilizado, sem restrições.

Assim, o ecrã da figura 43 não será mais apresentado quando, por exemplo, o sistema passar do "Modo de Segurança" para o "Modo Normal" e a etapa 4 ficar ativa. Neste caso, a variável "Ciclo de Aquecimento Efetuado" tem o valor 1, o que desativa a etapa 4 e ativa a etapa 13.
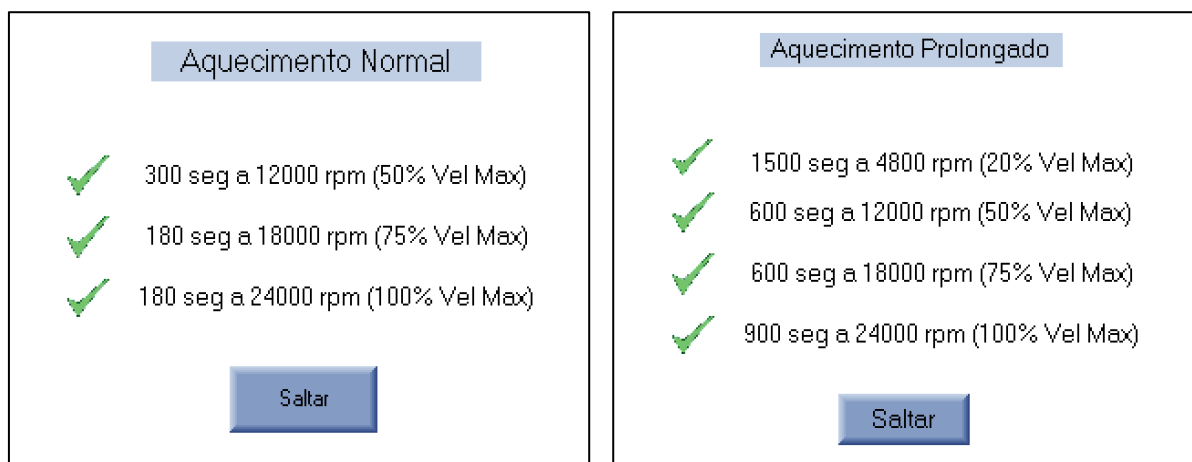
Figura 44 - Ecrãs apresentados durante a realização do ciclo de aquecimento (á direita no ciclo de aquecimento normal, à esquerda no ciclo de aquecimento prolongado)

O ciclo de aquecimento deverá ser repetido sempre que o equipamento for arrefecido até a temperatura ambiente. O *spindle* existente não possui o sensor RTD descrito no manual de utilização deste equipamento. Assim, não é possível comparar a temperatura do equipamento com a temperatura presente na sua envolvência. Para contornar esta limitação, foi programada a condição de que se o sistema estiver completamente imobilizado durante 30 minutos, o ecrã da figura 43 irá aparecer novamente e um novo ciclo de aquecimento deverá ser efetuado.

Após o ciclo de aquecimento estar concluído, o ecrã da figura 45 aparece na HMI.
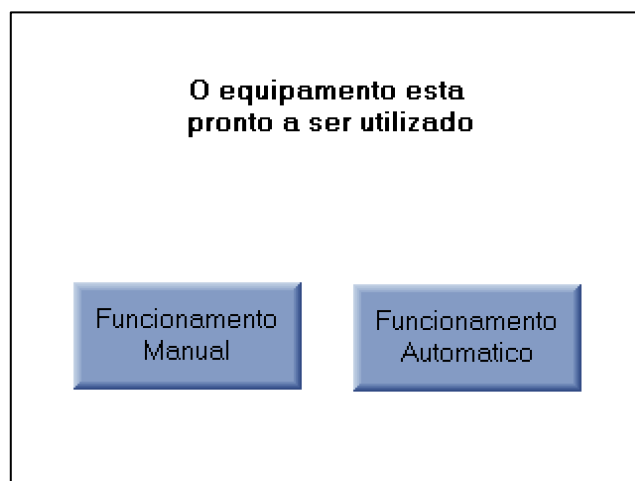
Figura 45 - Ecrã apresentado após a conclusão do ciclo de aquecimento

4.3.5 Implementação do Funcionamento Manual

O funcionamento manual do sistema é iniciado quando o botão "Funcionamento Manual" apresentado na figura 45 é atuado. Esta ação fará com que a etapa 14 seja ativa e que o ecrã da figura 47 surja na HMI.
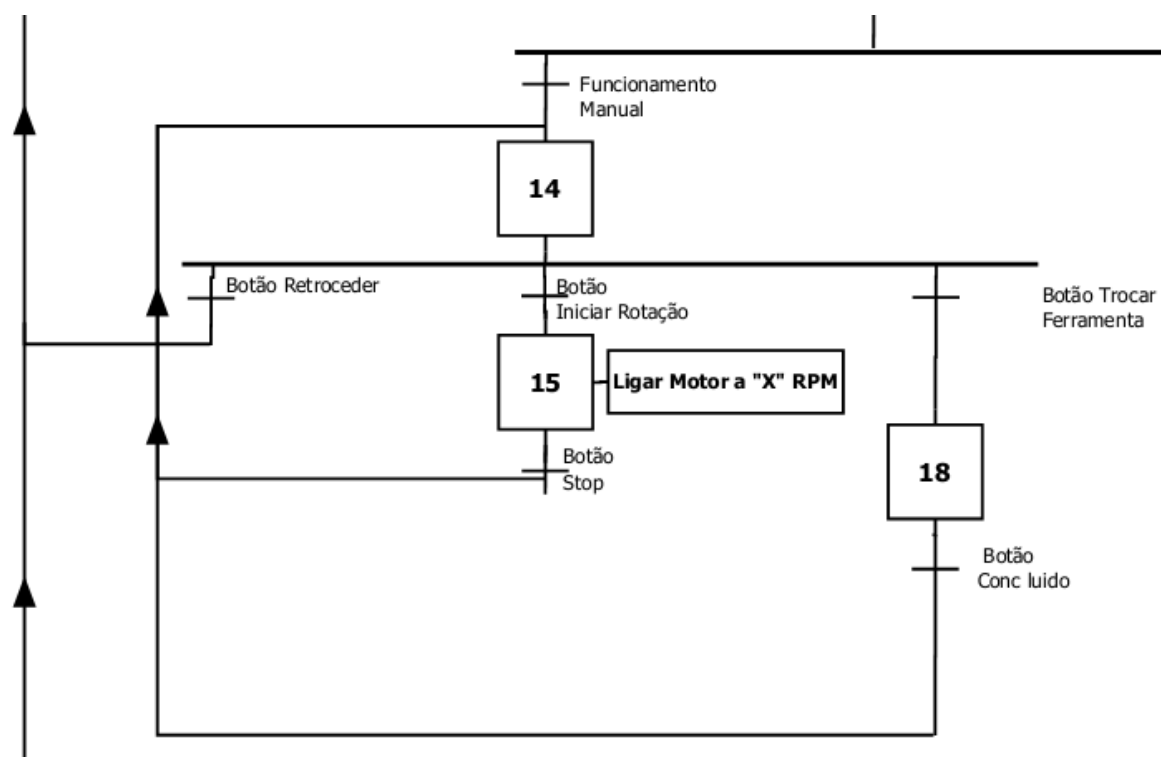
Figura 46 - Detalhe da parte de Funcionamento Manual do GRAFCET 2
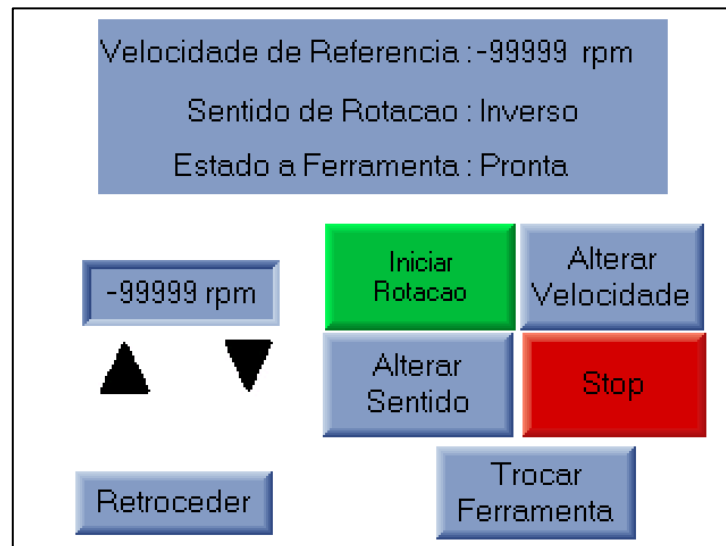
Figura 47 - Ecrã para o Funcionamento Manual do Sistema

Neste ecrã são apresentadas informações do estado atual do sistema, como o sentido de rotação, o estado da ferramenta e a velocidade de referência.

No ecrã da figura 47 existe um campo onde é possível selecionar a velocidade de referência desejada. Este valor pode ser alterado pela atuação das setas existentes, sendo que por cada atuação o valor é aumentado ou diminuído em 1000 unidades. Este valor poderia ser menor mas foi feito um equilíbrio entre os requisitos da aplicação e o número de vezes que seria necessário atuar as setas para atingir determinada velocidade. O valor que este campo apresenta está limitado ao intervalo entre 0 e 40000 rpm. O valor que aparece neste campo passará a ser a velocidade de referência quando for atuado o botão "Iniciar Rotação" ou "Alterar Velocidade".

Existem ainda outros dois botões para alteração do movimento do *spindle*. O botão "Alterar Sentido" permite ao utilizador mudar o sentido de rotação do *spindle*. A atuação do botão "Stop" pára o movimento do *spindle*.

Quando o estado da ferramenta é não pronto, nenhum dos botões de funcionamento ("Iniciar Rotação", "Alterar Sentido", "Stop" e "Alterar Velocidade") são visíveis. Assim, não é possível iniciar o movimento de rotação e é apenas possível aceder ao ecrã de mudança de ferramenta (figura 48) ou retroceder para o menu anterior (figura 45).

Ainda dentro do ecrã da figura 46, é possível aceder ao ecrã para mudança manual da ferramenta apresentado na figura 48 através do botão "Trocar Ferramenta".

Quando o *spindle* está parado, os botões "Alterar Velocidade", "Stop", e "Alterar Sentido" não se encontram visíveis. Por outro lado, os botões "Retroceder", "Troca Ferramenta" e "Iniciar Rotação" não estão visíveis quando o *spindle* está em funcionamento.
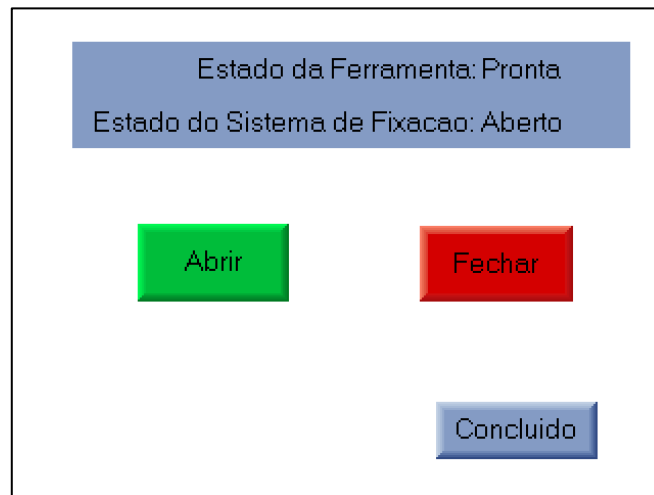
Figura 48 - Ecrã para a mudança de ferramenta manual

Neste ecrã são apresentadas algumas informações do estado atual do sistema, como o estado da ferramenta e o estado do sistema de fixação. Estas informações serão alteradas consoante as ações que forem tomadas.

Neste ecrã existem dois botões que controlam a atuação da electro válvula existente no interior do quadro elétrico e que controla o fluxo de ar para o sistema de fixação da ferramenta. Quando o botão “Abrir” é atuado, a válvula comuta e o sistema de fixação da ferramenta liberta a ferramenta. Se o botão “Fechar” for atuado, o fluxo de ar é interrompido e o sistema de fixação fechado.

O botão “Concluído” leva o sistema ao ecrã da figura 47.

4.3.6 Implementação da comunicação Autómato - Controlador IRC5

A ideia inicial para a implementação da comunicação entre o autómato e o controlador IRC5 era a utilização de uma ligação *Ethernet* (possivelmente *Fast Ethernet*) e do protocolo de comunicação TCP/IP.

Esta estratégia envolve a comunicação recorrendo a *sockets* entre o autómato e o controlador IRC5. Assim, foi programada no software *RobotStudio* uma rotina para teste desta comunicação. Quando foi feita a tentativa de implementação no controlador real, verificou-se que para utilização dos comandos essenciais para este tipo de comunicação, como o *Create Socket*, *Connect Socket*, entre outros, era necessário a opção “*PC Interface*” estar ativa no controlador IRC5 utilizado. Uma vez que esta opção não se encontrava ativa e que o custo da sua aquisição não era comportável, foi necessário encontrar uma alternativa para esta

comunicação entre o controlador do robô e o autómato. A solução encontrada passa pelo recurso a uma comunicação utilizando as entradas e saídas digitais disponíveis existentes no controlador.

Assim sendo, foram utilizadas 6 saídas digitais e 1 entrada digitais do controlador do robô. A função de cada uma das entradas e saídas é apresentada na tabela 9.

Para que o controlador do robô possa especificar ao autómato qual a velocidade de rotação do spindle, foi adotada uma estratégia de só usar uma linha de saída e gerar nessa saída um trem de impulsos proporcional à velocidade desejada.

A saída DO10_10 do controlador foi ligada a entrada 0 do autómato. Esta entrada pode ser configurada como "*high speed counter*". Assim, foi implementado um contador que irá contar o número de transições ascendentes e descentes que o sinal sofre durante 1 segundo a partir da primeira transição ascendente do sinal. Foram efetuados testes e verificado que o autómato consegue detetar pelo menos 20 transições ascendentes e descentes num segundo.

Tabela 9 - Condutores utilizados na comunicação Autómato-Controlador IRC5

Endereço Robô	**Endereço Autómato**	**Função**
DO10_9	Entrada 3	ON/OFF *Spindle*
DO10_10	Entrada 0	Velocidade de Rotação
DO10_11	Entrada 4	Ativar/Desativar *Spindle*
DO10_12	Entrada 5	Ativar/Desativar Troca Ferramenta
DO10_13	Entrada 6	Libertar/Prender Ferramenta
DI10_10	Saída 5	Linha de *Handshake*

Uma vez que o número de rotações máximo do *spindle* são 40000 rpm, cada transição detetada irá representar 40000/20 = 2000 rpm. Este valor será a resolução mínima de velocidade quando o sistema estiver em funcionamento automático. Seria possível descer o valor desta resolução mínima aumentando o número de pulsos enviados mas foi considerado que o valor de 2000 rpm é suficiente para a aplicação.

As outras entradas do autómato não necessitaram de ser configuradas como "*high speed counters*", uma vez que não necessitam de contar transições.

A saída 5 do autómato funciona como uma linha de *handshake*, indicando ao robô que a ordem que foi enviada anteriormente foi realizada com sucesso e que se encontra livre para

processar a próxima ordem. O controlador do robô espera sempre a receção do sinal de *handshake* antes de proceder á próxima instrução.

A saída DO10_11 do controlador IRC5 (ligada à entrada 4 do autómato) tem como função a ativação/desativação do *spindle*. A ativação do *spindle* é feita quando esta entrada toma o valor lógico 1 e o sistema está configurado para funcionamento automático. No caso de o *spindle* estar ativo, apenas podemos iniciar a rotação do *spindle* ou proceder à desativação do *spindle*, sendo que a operação de trocar ferramenta se encontra bloqueada nesta etapa. A desativação do *spindle* é feita colocando esta saída com o valor lógico 0.

A saída DO10_9 do controlador IRC5 (ligada à entrada 3 do autómato) tem como função ligar/desligar do *spindle*. Quando as saídas DO10_9 e DO10_11 têm o valor 1 e o sistema está configurado para funcionamento automático, o *spindle* inicia o seu funcionamento. Quando esta saída comuta para 0, o *spindle* pára a rotação.

A saída DO10_12 do controlador IRC5 (ligada à entrada 5 do autómato) tem como função a ativação/desativação da troca de ferramenta do *spindle*. A ativação da troca de ferramenta é feita quando esta entrada toma o valor lógico 1 e o sistema está configurado para funcionamento automático. Nesta situação apenas é possível abrir o sistema de fixação da ferramenta do *spindle* ou proceder à desativação da troca de ferramenta, sendo que a operação de iniciar a rotação do *spindle* se encontra bloqueada. A desativação da troca de ferramenta do *spindle* é feita colocando esta saída com o valor lógico 0.

A saída DO10_13 do controlador IRC5 (ligada à entrada 6 do autómato) tem como função abrir/fechar o sistema de fixação da ferramenta do *spindle*. Quando as saídas DO10_13 e DO10_12 têm o valor 1 e o sistema está configurado para funcionamento automático, a electroválvula existente dentro do armário elétrico abre, permitindo o fluxo de ar comprimido a 6 bar para o interior do *spindle*. O ar comprimido irá comprimir a mola do sistema de fixação, permitindo que a ferramenta fique solta. Quando a operação de mudança de ferramenta está concluída, esta toma o valor lógico 0, quer seja para agarrar uma nova ferramenta ou para fechar o sistema de fixação sem nenhuma ferramenta montada.

Esta sequência de operações aqui apresentada será melhor analisada em conjunto com as respetivas rotinas de programação no subcapítulo seguinte.

4.3.7 Implementação do Funcionamento Automático

O Funcionamento Automático diz respeito ao funcionamento do *spindle* quando este é controlado apenas pelo controlador IRC5. A comunicação entre estes dois equipamentos foi descrita em termos de *hardware* no subcapítulo anterior. Aqui é referida a implementação de *software* que foi desenvolvida.

A parte do GRAFCET 2 que diz respeito ao funcionamento automático é apresentado na figura 49. Aqui podemos ver que existem 2 ações distintas que podem ser executadas quando este modo está ativo: a operação de Ligar/Desligar o *Spindle* e a operação de troca de ferramenta.
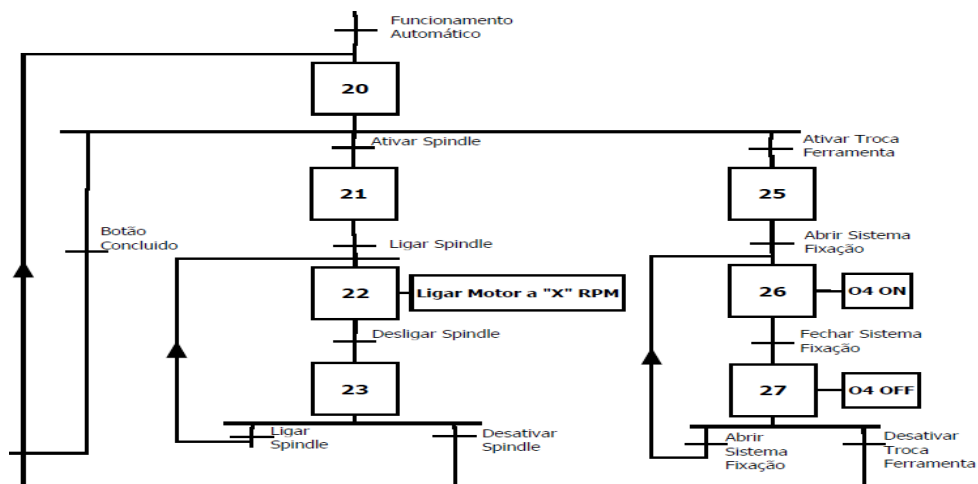
Figura 49 - Detalhe da parte de Funcionamento Automático do GRAFCET 2

O Funcionamento Automático é ativado quando é atuado o botão "Funcionamento Automático" presente no ecrã da figura 45. Quando ocorre esta ação, surge na HMI o ecrã apresentado na figura 50.
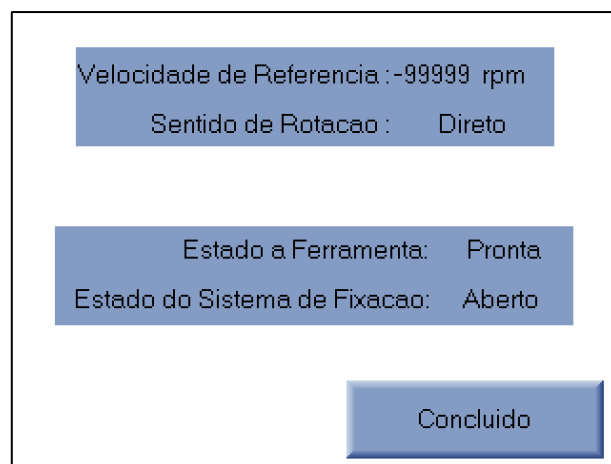
Figura 50 - Ecrã visível durante o Funcionamento Automático do Sistema

Aqui são apresentadas algumas informações do estado atual do sistema como a velocidade de referência, o sentido de rotação, o estado da ferramenta e do sistema de fixação desta.
O botão concluído está apenas visível quando não está a ser feita nenhuma operação no equipamento, ou seja, quando o GRAFCET tem a etapa 20 ativa. A atuação deste botão faz surgir de novo na HMI o ecrã da figura 45.

Para o controlo das ações que o sistema pode realizar foram desenvolvidas rotinas que podem ser incorporadas num programa RAPID executado pelo controlador do robô. Foram desenvolvidas 11 rotinas com diferentes funções:

- ativar_spindle
- desativar_spindle
- ligar_spindle
- desligar_spindle
- set_speed
- ativar_troca_ferramenta
- libertar_ferramenta
- prender_ferramenta
- ativar_troca_ferramenta
- sentido_inverso
- sentido_inverso

O código de programação destas rotinas pode ser analisado no anexo C. A titulo de exemplo, o código RAPID da rotina ativar_spindle é apresentado na figura 51.

```
39  PROC ativar_spindle()
40      Set DO10_11_ativar_spindle;
41      WaitDI DI10_9_handshake,1;
42  ENDPROC
```

Figura 51 - Rotina ativar_spindle

Esta rotina é bastante simples. A primeira linha de código faz o set à saída DO10_11_ativar_spindle, enquanto a segunda coloca o programa em espera até que a entrada DI10_9_handshake tenha o valor 1. Esta linha dá a indicação que o autómato recebeu corretamente a indicação que lhe foi enviada.

As rotinas desativar_spindle, desligar_spindle, ativar_troca_ferramenta, desativar_troca_ferramenta, libertar_ferramenta e prender_ferramenta têm um código e modo de funcionamento muito semelhante a esta, sendo que apenas é alterada a variável a que a rotina refere e, em algumas delas, a operação de set é substituída por reset. Por este motivo, estas rotinas não são abordadas em detalhe neste capítulo mas são apresentadas no anexo C.

O código RAPID da rotina set_speed é apresentado na figura 52:

```
15  PROC set_speed(num x)
16      Var num pulsos;
17      Var num index:=0;
18      Var num setenta:=70;
19      Var num tempo:=1.1;
20      pulsos := x Div setenta;
21      While index < pulsos Do
22          Set DO10_10_set_speed;
23          WaitTime 0.05;
24          tempo:= tempo - 0.05;
25          index:=index+1;
26          if index < pulsos then
27          Reset DO10_10_set_speed;
28          WaitTime 0.05;
29          tempo:= tempo - 0.05;
30          index:=index+1;
31          ENDIF
32      ENDWHILE
33      WaitTime tempo;
34      Reset DO10_10_set_speed;
35      WaitDI DI10_9_handshake,1;
36      WaitTime 0.2;
37  ENDPROC
```

Figura 52 - Rotina set_speed

Após declarar as variáveis necessárias, o numero indicado dentro de parenteses é dividido pela constante 70. Este valor será igual ao número de transições ascendentes e descendentes que o controlador deverá enviar ao autómato.

Para o envio do número de pulsos correto foi usada uma instrução *While*. Enquanto a variável índex for inferior ao número de transições, é feito o set da variável DO10_10_set_speed e adicionada uma unidade ao valor índex. Após isto, se o índex for menor que o número de transições, é feito o reset da variável DO10_10_set_speed e adicionada outra unidade ao valor índex. Este ciclo é repetido até que o índex seja igual ao número de transições.

Quando isto acontece o sistema espera um tempo igual à variável “tempo”, valor necessário para completar 1,1 segundos desde que a primeira transição ocorreu. O autómato está programado para contar o número de transições ascendentes e descendentes durante 1 segundo após a primeira transição ascendente e é dada uma margem de segurança de 0.1s.

Após este tempo é feito o reset da saída "DO10_10_set_speed" e o programa espera até que a entrada "DI10_9_handshake" tenha o valor 1. Quando esta variável toma o valor lógico 1, é aguardado um tempo de 0.2 segundo e a rotina é terminada.

A figura 53 apresenta um esquema da evolução temporal da saída ativa nesta rotina.
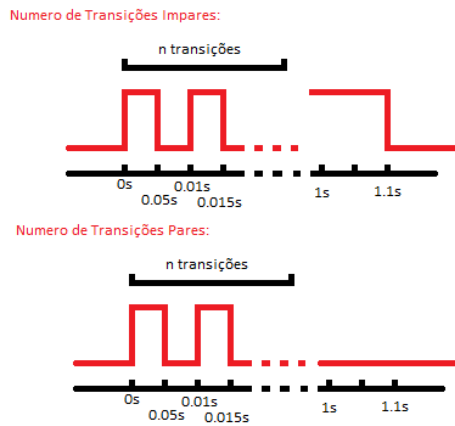
Figura 53 - Esquema da contagem do número de transições da saída DO10_19_set_speed

O código RAPID da rotina ligar_spindle é apresentado na figura 54:

```
44  PROC ligar_spindle (num speed)
45      set_speed (speed);
46      Set DO10_9_on_off_spindle;
47      WaitDI DI10_9_handshake,1;
48      WaitTime 0.2;
49  ENDPROC
```

Figura 54 - Rotina ligar_spindle

Esta rotina utiliza a rotina set_speed para o envio da velocidade indicada para o autómato. Após a conclusão da rotina set_speed, é feito o set à saída DO10_9_on_off_spindle e o programa espera até que a entrada "DI10_9_handshake" tenha o valor lógico 1. Quando esta variável tem o valor 1, é aguardado um tempo de 0.2 segundo e a rotina é terminada.

Foram também elaboradas rotinas que permitem definir o sentido de rotação da ferramenta e que poderão ser utilizadas em qualquer ponto do programa principal, mesmo com o *spindle* parado. Estas rotinas são apresentadas na figura 55 e na figura 56

```
82  Proc sentido_inverso()
83      Set DO10_14_sentido_rotacao_spindle;
84      WaitTime 0.05;
85      Reset DO10_14_sentido_rotacao_spindle;
86      WaitDI DI10_9_handshake,1;
87      WaitTime 0.2;
88  ENDPROC
```

Figura 55 - Rotina sentido_inverso

```
90  Proc sentido_direto()
91      Set DO10_14_sentido_rotacao_spindle;
92      WaitTime 0.05;
93      Reset DO10_14_sentido_rotacao_spindle;
94      WaitTime 0.05;
95      Set DO10_14_sentido_rotacao_spindle;
96      WaitTime 0.05;
97      Reset DO10_14_sentido_rotacao_spindle;
98      WaitDI DI10_9_handshake,1;
99      WaitTime 0.2;
100 ENDPROC
```

Figura 56 - Rotina sentido_direto

Ambas as rotinas utilizam a saída "DO10_14_sentido_rotacao_spindle" para enviar ao autómato qual o sentido de rotação desejado. Na rotina sentido_inverso é enviado um pulso ao autómato. Na rotina sentido_direto são enviados dois pulsos. Após enviado o número de pulsos desejado, o programa espera até que a entrada "DI10_9_handshake" tenha o valor 1. A figura 54 apresenta a evolução do sinal "DO10_14_sentido_rotacao_spindle" dependendo da rotina desejada.
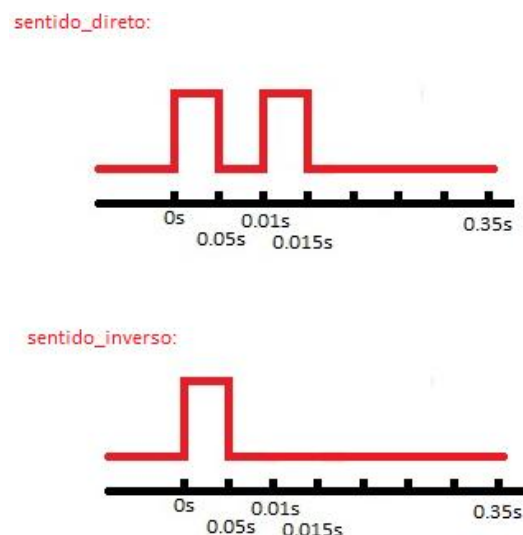
Figura 57 - Evolução da saída "DO10_14_sentido_rotacao_spindle" dependendo da rotina utilizada

Todas as rotinas anteriormente apresentadas devem ser copiadas para o módulo do programa principal que o robô irá seguir quando a utilização do spindle for necessária.

Se a resposta do autómato através da linha de *handshake* não funcionar, o programa RAPID ficará bloqueado na primeira instrução enviada ao autómato, que deverá ser ativar spindle ou ativar a troca de ferramenta. Neste caso é recomendado que se verifique se o autómato se encontra em funcionamento ou se existe algum problema com a cablagem. Após corrigido o problema, o programa RAPID do controlador do robô deve ser reiniciado, por forma a evitar maus funcionamentos do sistema.

A figura 58 apresenta um exemplo da possível utilização destas rotinas num programa RAPID de programação de um robô.

```
30 PROC main()
31     MoveJ Target_10,v1000,z100,Tooldata_1\WObj:=wobj0;
32     MoveL Target_40_2,v1000,z100,Tooldata_1\WObj:=Ferramenta_1;
33     ativar_spindle;
34     sentido_direto;
35     ligar_spindle (20000);
36     MoveJ Target_10,v1000,z100,Tooldata_1\WObj:=wobj0;
37     MoveL Target_40_2,v1000,z100,Tooldata_1\WObj:=Ferramenta_1;
38     sentido_inverso;
39     set_speed(25000);
40     MoveL Target_90_2,v50,fine,Tooldata_1\WObj:=Ferramenta_1;
41     MoveL Target_140_2,v50,fine,Tooldata_1\WObj:=Ferramenta_1;
42     desligar_spindle;
43     desativar_spindle;
44
45     MoveJ Target_10,v1000,z100,Tooldata_1\WObj:=wobj0;
46     ativar_troca_ferramenta;
47     libertar_ferramenta;
48     MoveL Target_90_2_2,v50,fine,Tooldata_1\WObj:=Ferramenta_1;
49     prender_ferramenta;
50     desativar_troca_ferramenta;
51 ENDPROC
```

Figura 58- Exemplo de aplicação das rotinas desenvolvidas

4.4 Testes do Sistema

Após a montagem de todos os componentes, programação do autómato e implementação de todas as comunicações, foram realizados os testes finais do sistema implementado. As figuras 59 e 60 mostram o aspeto final do *spindle* montado no robô ABB IRB2400 durante estes testes.

Estes testes foram concluídos com sucesso, uma vez que o sistema final cumpre com todos os requisitos previamente especificados. O funcionamento manual e o funcionamento automático foram testados e funcionam como esperado. Uma das limitações encontradas e que já era esperada, foi o reduzido comprimento dos cabos de ligação do armário elétrico ao *spindle*, como é visível na figura 60. Este fato poderá limitar alguns deslocamentos que o robô possa efetuar quando o *spindle* está em funcionamento
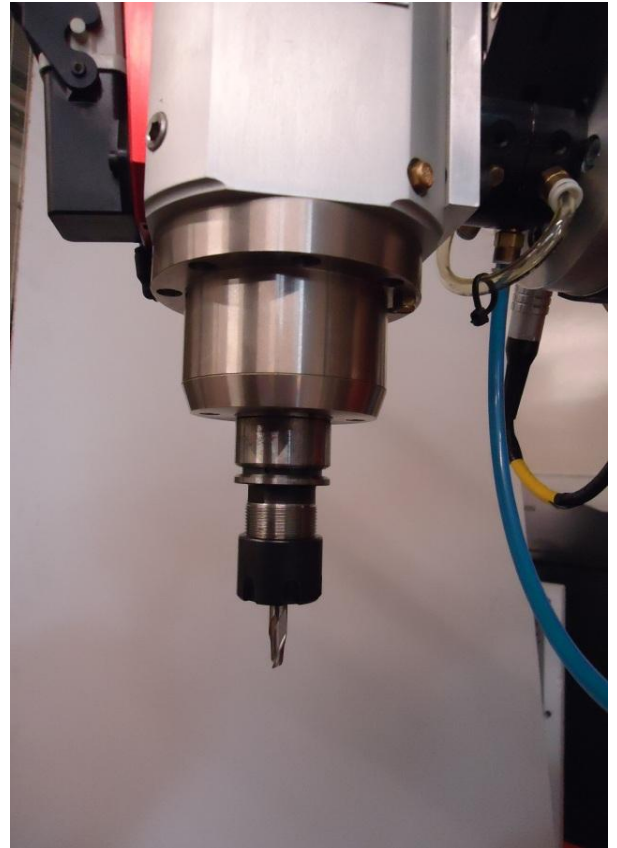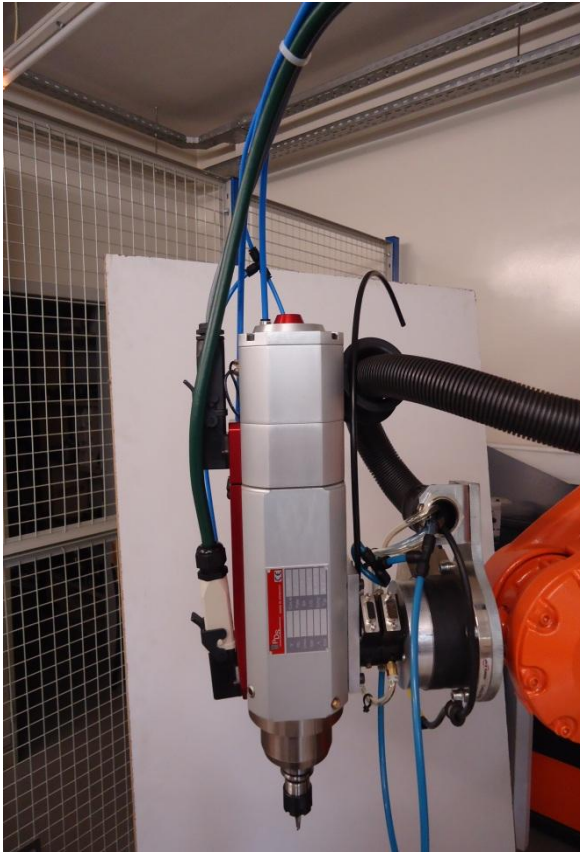
Figura 59 - Aspeto final do *spindle* montado no robô ABB IRB2400
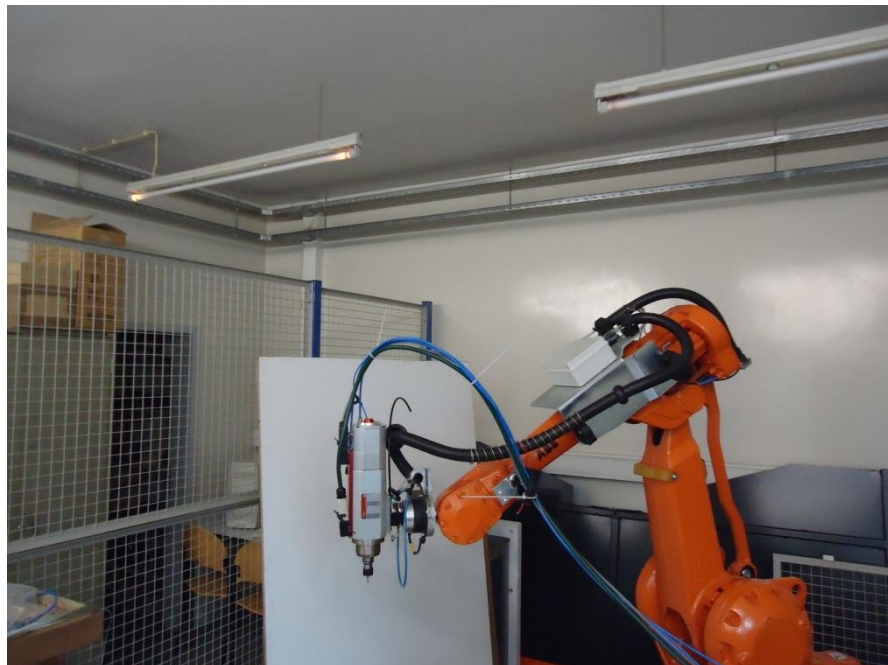
Figura 60 - *Spindle* montado no robô ABB IRB 2400

Capítulo 5
Conclusões e Trabalhos Futuros

5.1 Conclusões

A utilização de robôs industriais para realização de algumas operações de maquinagem onde os requisitos de precisão dimensional da peça não sejam muito elevados tem vindo a ganhar cada vez mais importância no mundo industrial uma vez que os robôs são bastante adaptáveis e flexíveis, fáceis de programar e possuem um custo relativamente baixo. A maquinagem usando robôs industriais necessita que o robô esteja equipado com um *spindle* para fornecer a ferramenta de corte o movimento de rotação necessário.

Como relatado ao longo deste relatório, este projeto passou pela instalação e implementação de um *spindle* num robô industrial ABB IRB2400. A arquitetura de controlo do sistema a implementado passa pela utilização de um autómato programável que comunica com um variador de frequência através de uma ligação série RS485 e com o controlador do robô através de 7 linhas I/O dedicadas. Com este sistema pode controlar-se de a velocidade e sentido de rotação do *spindle* e a abertura e fecho do sistema automático de mudança de ferramenta.

O sistema pode ser controlado manualmente pela consola HMI integrada ou pelo controlador IRC5 instalado. O sistema final apresentado cumpre os requisitos funcionais que tinham sido inicialmente propostos, sendo assegurados os cuidados indispensáveis à segurança da utilização do equipamento. Há que referir que, por indisponibilidade de *hardware*, foi necessário implementar a comunicação autómato-controlador do robô por meio de ligação via linhas de I/O dedicadas, substituto da ligação *Ethernet*.

A principal dificuldade sentida, durante a realização deste trabalho, prendeu-se com a implementação da comunicação autómato-variador de frequência por uma ligação RS485. O

manual de utilização do autómato não possui informações detalhadas de como configurar e implementar estas comunicações. Este facto levou a vários dias despendidos à procura de correto funcionamento da comunicação entre os dispositivos. O manual de utilização do autómato não indica que este equipamento não suporta MODBUS ASCII, o que levou a perda de mais algumas horas a tentar implementar este modo.

Este projeto envolveu a compra de diverso material pneumático, elétrico, a montagem de um quadro elétrico, bem como a realização de cabos elétricos. O tempo de espera por algum deste material foi bastante longo, em particular no caso do material pneumático. Este fato também levou a alguns períodos menos produtivos durante a fase de montagem dos componentes.

Apesar destes contratempos, foi possível cumprir os objetivos propostos e ter um sistema funcional.

A realização desta dissertação permitiu uma aprendizagem importante em diversas áreas da engenharia, nomeadamente:

- Seleção de material elétrico e pneumático para que um sistema cumpra determinados requisitos funcionais e de segurança;
- Conceção de uma arquitetura de controlo com a integração de equipamentos diversos, comunicando entre eles por diferentes tipos de ligações;
- Configuração de diversos tipos de comunicações;
- Conceção de um sistema pneumático para cumprir os requisitos do sistema de refrigeração e do sistema automático de mudança de ferramenta requeridos pelo equipamento;
- Programação de um autómato num sistema com alguma dimensão e complexidade.

5.2 Trabalhos Futuros

No final deste trabalho, passa a ser possível realizar pequenos trabalhos de maquinagem de materiais plásticos, espumas e alguns metais, entre outros, recorrendo ao sistema correntemente implementado. Apesar de o sistema estar funcional e pronto a ser utilizado, existem ainda alguns melhoramentos que podem ser implementados.

Assim, como trabalhos futuros sugere-se a implementação da ligação *Ethernet* entre o controlador do robô e o autómato. Esta comunicação iria permitir aumentar a resolução de velocidade no funcionamento automático e iria eliminar as sete linhas de I/O dedicadas instaladas, que seriam substituídas por um cabo *Ethernet*.

A monitorização remota das condições de operação do spindle durante a operação de maquinagem (registo da velocidade, corrente consumida, tensão fornecida, forças e binários, entre outros) seria também um melhoramento considerável ao sistema atual. A implementação da ligação *Ethernet* entre o controlador do robô e o autómato iria simplificar bastante a monitorização do sistema.

A construção de uma nova cablagem, de maior extensão do que a atual, seria também importante para a utilização do robô com menos limitações nas duas deslocações.

Referências

[1] http://www.citi.pt/educacao_final/trab_final_inteligencia_artificial/robotica.html, acesso em 2012/06/14.

[2] http://olhomagic.blogspot.com/2011/03/de-onde-vem-palavra-robo.html, acesso em 2012/06/14.

[3] KOREN, YORAM, “Robotics for Engineers”, McGraw-Hill, 1985.

[4] ISO, “ISO Standard 8373: 1994 – Manipulating Industrial Robots”, 1994.

[5] ZHANG, HUI et al, “Machining with Flexible Manipulator: Toward Improving Robotic Machining Performance”, International Conference on Advanced Intelligent Mechatronics, 2005.

[6] WORLD ROBOTICS, “Executive Summary of World Robotics 2011 Industrial Robots and Service Robots”, 2011.

[7] http://spectrum.ieee.org/robotics/industrial-robots/the-rise-of-the-machines/0, acesso em 2012/06/14.

[8] ABELE, E.; WEIGOLD, M.; ROTHENBÜCHER, S.; “Modeling and Identification of an Industrial Robot for Machining Applications”, Institute of Production Management, Darmstadt University of Technology, 2007.

[9] http://www.rodin4d.com/en/Products/manufacturing/robots. Acesso em 2012/06/14.

[10] JUN, MARTIN B. et al; “Evaluation of a spindle-based force sensor for monitoring and fault diagnosis of machining operations”, Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign, 2001.

[11] DYNOMAX, “The Book of spindles Part 1”. Disponível em: http://pt.scribd.com/doc/85229016/35/Corporate-Overview

[12] HSD, “Installation, operation and maintenance manual of Electrical Spindles”, Electro Spindle Department of the technical office at HSD S.p.a. Disponível em: www.eagleeyecnc.com/manual/hsd_manual.pdf

[13] www.larkencnc.com/spindles.htm . Acesso em 2012/06/14

[14] http://www.pdscolombo.com/prod_gcstone.php. Acesso em 2012/06/14.

[16] BORGES, FÁTIMA, "O que é um PLC (autómato)?", Centro de Formação Schneider Electric, 2008.

[15] PIRES, J.NORBERTO, "Automação Industrial", Lidel – edições técnicas, 2002.

[17] MAGALHÃES, ANTÓNIO PESSOA, Apontamentos da unidade curricular de Computação Industrial, Faculdade de Engenharia da Universidade do Porto, 2011.

[18] http://pplware.sapo.pt/networking/redes-sabe-o-que-e-o-modelo-osi/. Acesso em 2012/06/14.

[19] http://www.interfacebus.com/Design_Connector_RS422.html. Acesso em 2012/06/14.

[20] KRON MEDIDORES, "Conceitos Básicos de RS 485 e RS422", Kron. Disponível em http://www.kronweb.com.br .

[21] PERRIN, BOB, "The Art and Science of RS-485", Circuit Cellar Online, 1999.

[22] SPURGEON, CHARLES E., "*Ethernet*: The definitive Guide", O'Reilly & Associates, Inc, 2000.

[23] http://en.wikipedia.org/wiki/Carrier_sense_multiple_access_with_collision_detection. Acesso em 2012/06/14

[24]http://www.technicalhowto.com/Networking/Juniper/Certifications/JNCIA/nf/networking_fundamentals.html. Acesso em 2012/06/14

[25] http://www.simplymodbus.ca/FAQ.htm. Acesso em 2012/06/14

[26] ACROMAG INCORPORATED, "Introduction to Modbus TCP/IP", Acromag, Inc, 2005.

[27] MODICON, "Modicon Modbus Protocol Reference Guide", Modicon, Inc., Industrial Automation Systems, Revisão J, Junho 1996.

[28] http://www.kiming.co.kr/bbs/bbs/board.php?bo_table=pds&wr_id=30. Acesso em 2012/06/14.

[29]http://www.abb.com/product/seitp327/657d58e39c804f64c1256efc002860a7.aspx?productLanguage=pt&country=PT. Acesso em 2012/06/14.

[30] MARSHALL, PETER S., RINALDI, JOHN S., "Industrial *Ethernet*", ISA – The Instrumentation, Systems, and Automation Society, segunda edição, 2005.

[31] http://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.3/suselinux-adminguide_en/cha.basicnet.html. Acesso em 2012/06/14.

[32] http://www.gta.ufrj.br/grad/06_1/ipv6/. Acesso em 2012/06/14.

[33] http://www.clubedohardware.com.br/printpage/Como-o-Protocolo-TCP-IP-Funciona-Parte-1/1351. Acesso em 2012/06/14.

[34] http://under-linux.org/albums/magnun/imagem-para-posts-2-146/7237-4-cabecalho-tcp/. Acesso em 2012/06/14.

[35] http://www.novell.com/documentation/suse91/suselinux-adminguide/html/ch14.html. Acesso em 2012/06/14.

[36] PDS – PRECISE DRIVE SYSTEMS, "Operation Manual Spindle XLC/XLA 070", PDS, Dallas.

[37] http://www.unitronics.com/Series.aspx?page=Vision350. Acesso em 2012/06/14.

[38] http://www.wolfautomation.com/assets/15/VFD-VE1mi.jpg. Acesso em 2012/06/14.

[29]http://www.delta.com.tw/product/em/drive/ac_motor/ac_motor_product.asp?pid=1&cid=1&itid=9. Acesso em 2012/06/14 .

[40] DELTA ELECTRONICS, "VFD_VE User Manual", Delta Electronics, Taiwan. Disponível em http://www.delta.com.tw/ .

Bibliografia

MARTINS, IVO, “Introdução aos Autómatos Programáveis”, Texto de apoio às aulas, Instituto Superior de Engenharia, Universidade do Algarve, 2010.

SILVIO, MARIANO; GASPAR, PEDRO; Texto de Apoio às Aulas de Automação Industrial, Universidade da Beira Interior, Portugal, 2010.

GOLDIE, JOHN, “Ten ways to bulletproof RS485 Interfaces”, Texas Instruments, Outubro 1996.

Anexo A
Esquema Elétrico Implementado

SCHOOL LICENCE
Sinal S1
Sinal S2
Sinal S3
Sinal S4
3.5
3.5
3.5
3.5
Sinal Thermal Switch
2.6
5.0
0 V (4)
DI
-I 02
Sinal Sensor 1
Shaft Rotation
DI
-I 08
Sinal Sensor 2
Tool Expulsion
DI
-I 09
Sinal Sensor 3
Tool Clamped
DI
-I 10
Sinal Sensor 4
Cylinder Returned
DI
-I 11
Sinal Thermal Switch
Sinal Thermal Switch
Sinal Pres 1
7.0
Sinal Pres 2
7.0
2.1
Erro VF
2.2
Emergencia
DI
-I 12
Sinal Botao Emergencia
DI
-I 13
Sinal Presostato 6 bar
DI
-I 14
Sinal Presostato 2 bar
DI
-I 15
Erro Variador Freq.
Entradas PLC
SCHOOL LICENCE
Logo
Esquema de Entradas do PLC
Project:
Esquema Electrico
Date:
25-02-2012
Drawing no.:
Function:
Init.:
Location:
Rev.:
Total sheets:
7
Sheet:
4
Next sheet:
5

Saidas PLC
SCHOOL LICENCE
- Logo -
Esquema de Saidas do PLC
-O 06
Sinal Valvula 1
Fornecimento de Ar
-O 04
Sinal Valvula 2
Soltar Ferramenta
-O 02
Sinal ON Var. Freq.
Ligar Var. Freq.
-O 03
Sinal OFF Var. Freq.
Desligar Var. Freq.
-O 05
Sinal Handshake
para o Robo
24V (3)
0 V (3)
Sinal VF
Sinal VP2
Sinal VP1
Robo Handshake
Project:
Esquema Electrico
Date:
01-03-2012
Drawing no.:
Function:
Location:
Rev.:
Total sheets:
7
Sheet:
6
Next sheet:
7
SCHOOL LICENCE

SCHOOL LICENCE
Entradas PLC
Logo
7.3 Sinal alterar_sentido_auto
7.3 Sinal on/off_spindle
7.3 Sinal libertar/prender_ferrame
7.3 Sinal ativar_troca_ferramenta
7.3 Sinal ativar_spindle
7.3 Sinal set_spindle_speed
4.0 0 V (4)
DI
I 00
Sinal set_spindle
DI
I 04
Sinal ativar_spindle
DI
I05
Sinal Ativar_troca_ferramenta
DI
I06
Sinal Libertar/Prender Ferramenta
DI
I 07
Sinal On/OFF Spindle
DI
I 16
Sinal alterar_sentido_rotacao
Esquema de Entradas do PLC
Project: Esquema Electrico
Date: 26-06-2012
Drawing no.:
Function:
Init.:
Location:
Rev.:
Total sheets: 7
Sheet: 5
Next sheet: 6

SCHOOL LICENCE
- Logo -
Presostato 1
6 bar
Presostato 2
2 bar
0 V (5)
24V (4)
Sinal Pres 1
Sinal Pres 2
Sinal VP1
Sinal VP2
0 V (7)
0 V (8)
-VP1
-VP2
PLC
0 V (6)
24V (5)
RS485
Ethernet
Controlador ABB
Sinal set_spindle_speed
Sinal ativar_spindle
Sinal ativar_troca_ferramenta
Sinal libertar/prender_ferrame
Sinal on/off_spindle
Sinal alterar_sentido_auto
Robo Handshake
EnergRobot2
EnergRobot1
Sistema de Arrefecimento e Outros
Project: Esquema Electrico
Date: 25-02-2012
Drawing no.:
Function:
Location:
Init.:
Rev.:
Total sheets: 7
Sheet: 7
Next sheet:
SCHOOL LICENCE

Anexo B

Circuito Pneumático Implementado

Tool Actuator
Spindle cooling and pressurization (6mm, 6 bar)
Spindle Cone Cleaning (4mm, 2 bar)
Spindle pressurization (4mm, 2bar)
Pressure Sensor 2
0
2 bar
Flow: 110 l/min @ 2 bar
Pressure Sensor 1
0
6 bar
Air Drier
Filter & Drain

Lista de material pneumático			
Requisitos de ar comprimido para spindle com sistema de mudança automática de ferramenta			
ISO 8573.1 class 1 para partículas e óleo (max. Particle size 0.1 micron, oil 0.01 mg/m3)			
ISO 8573.1 class 3 para humidade (dew point @ atmospheric pressure -22ºC)			
consumo 1100 l/min @ patm			
QT	**Equipamento**	**Função**	**Ref.**
1	Inline Desiccant dryer, 1/2	Tratamento do ar	1
1	Válvula Manual Shut Off G3/8	Unidade de tratamento do ar	1.1
1	Filtro ar, 1 micron com sistema purga ,3/8"	Unidade de tratamento do ar	1.2
1	Filtro ar/óleo (Coalescing filter) 0.01 micron, 3/8	Unidade de tratamento do ar	1.3
1	Válvula Redutora de pressão com manómetro 1/4	Unidade de tratamento do ar	1.4
2	Pressostato - Interruptor de pressão ajustável ,1-10 bar, G1/8	Pressostato	2
1	Eletroválvula distribuidora monoestável 3/2 NC solenóide 24Vdc, G1/4 (caudal 1100 Nl/min)	Válvula Direcional (air on/off)	3
1	Eletroválvula distribuidora monoestável 3/2 NC solenóide 24Vdc, G1/8	Válvula Direcional (tool unlock)	4
1	Válvula Redutora de pressão com manómetro 1/4	Válvula manométrica	5
2	Raccord - conector unid. trat. ar (válvula shutoff 3/8) / rede, com engate rápido para tubo 6mm	Raccord	
1	Raccord – conector para unid trat. Ar (válvula redutora de pressão 1/4)/ rede, com engate rápido para tubo 6mm	Raccord	
1	Raccord – conector para válvula redutora pressão (1/4)/ rede, com engate rápido para tubo 6mm	Raccord	
1	Raccord – conector para válvula redutora pressão (1/4)/ rede, com engate rápido para tubo 4mm	Raccord	
1	Raccord de União Passamuros para tubo 4mm/4mm , engates rápidos	Raccord	
3	Raccord de União Passamuros para tubo 6mm/6mm, engates rápidos	Raccord	
1	Raccord - conector para pressotato (1/8)/rede, com engate rápido para tubo 6mm	Raccord	
1	Raccord - conector para pressotato (1/8)/rede, com engate rápido para tubo 4mm	Raccord	
3	Raccord- conector pneumático em T, engate rápido, tubo 6mm	Raccord	
2	Raccord- conector pneumático em T, engate rápido, tubo 4 mm	Raccord	
3	Raccord - conector para válvula distribuidora (1/8)/ rede, com engate rápido para tubo 6mm	Raccord	
3	Raccord - conector para válvula distribuidora (1/4)/ rede, com engate rápido para tubo 6mm	Raccord	
1	Tubo pneumático PVC, diam. 6 mm, (30metros)	Tubo pneumático	
1	Tubo pneumático PVC, diam. 4 mm, (15metros)	Tubo pneumático	

Anexo C

Rotinas RAPID Desenvolvidas

```
MODULE Module1

PROC main()
        WaitTime 1;
        ativar_spindle;
        ligar_spindle;
        WaitTime 10;
        desligar_spindle;
        desativar_spindle;
        WaitTime 5;
        ativar_troca_ferramenta;
        libertar_ferramenta;
        prender_ferramenta;
        desativar_troca_ferramenta;
ENDPROC
PROC ativar_spindle()
        Set DO10_11_ativar_spindle;
        WaitTime 1;
ENDPROC
PROC ligar_spindle ()
        Set DO10_9_on_off_spindle;
        WaitTime 1;
ENDPROC
Proc desligar_spindle ()
        Reset DO10_9_on_off_spindle;
        WaitTime 1;
ENDPROC
Proc desativar_spindle ()
        Reset DO10_11_ativar_spindle;
        WaitTime 1;
ENDPROC
Proc ativar_troca_ferramenta ()
        Set DO10_12_troca_ferramenta_spindle;
        WaitTime 1;
ENDPROC
Proc libertar_ferramenta ()
        Set DO10_13_libertar_ferramenta;
        WaitTime 1;
ENDPROC
Proc prender_ferramenta ()
        Reset DO10_13_libertar_ferramenta;
        WaitTime 1;
ENDPROC
Proc desativar_troca_ferramenta ()
        Reset DO10_12_troca_ferramenta_spindle;
        WaitTime 1;
ENDPROC

ENDMODULE
```